Invertible secret image sharing for gray level and dithered cover images

Mustafa Ulutas, Güzin Ulutas, Vasif V. Nabiye
Department of Computer Engineering, Karadeniz Technical University, Trabzon 61080, Turkey

Abstract
Secret image sharing approaches have been extended to obtain covers from stego images after the revealing procedure. Lin et al.’s work in 2009 uses modulus operator to decrease the share image distortion while providing recovery of original covers. Their work use gray level or binary image as cover. Stego images have approximately 43 dB and 38 dB PSNR for gray level and binary covers respectively. Lin et al.’s work in 2010 provides enhanced embedding capacity but does not support binary covers. Gray level covers’ PSNR is reported approximately 40 dB. The proposed method enhances the visual quality of stego images regardless of intensity range of covers. Exploiting Modification Direction method is used to hide the shared values into covers. The method also utilizes modulus operator to recover original cover pixels. Stego image PSNR is approximately 47 dB for gray level covers. The method provides 4–7 dB increase respectively on the stego image quality compared to others. Stego images have also higher PSNR (43 dB) for dithered covers. The proposed method generates stego images with higher PSNR regardless of the intensity range of the cover image.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction
The development of network technologies makes the digital distribution of data more convenient. However, distributing the important data over public network makes the data vulnerable to attacks. Therefore, protection of digital data becomes a critical issue in recent years. Some techniques have been developed to overcome this problem. Cryptography and steganography are the two techniques used for protecting data. Former technique uses mathematical transformation to protect the data from the malicious eyes. Secret data is transformed into unreadable form to provide the security. However, latter method hides the data into a cover medium. This medium can be any content such as image, video, audio or a text file. Embedding the secret data into the least significant bits (LSB) of the cover medium is the most frequently used technique known as the LSB embedding in the literature.

Both of these methods accommodate the secret data into one medium. Corruption of the medium results in the loss of the secret data. Secret Sharing Schemes are used to overcome this problem. Shamir (1979) proposed a secret sharing method called (k, n) threshold secret sharing scheme. His method divides the secret among n participants. Each participant gets a piece of secret called share. If any k or more participants gather their shares, secret will be revealed. Otherwise, no information about the secret image is revealed. Shamir’s method uses Lagrange’s interpolation technique to reconstruct the secret. A (k − 1)th degree polynomial, f(x), with constant term as the secret is constructed. Evaluation of the polynomial f(x_i) for unique values of x_i constitutes shares. Therefore, shares are unique points on the polynomial f(x). Lagrange’s interpolation technique is used for reconstructing the secret from k points gathered from participants.

Blakley (1979) proposed another technique in the same year to share a secret. His method is based on geometric approach. Secret is a point in k-dimensional space according to his method. Shares are constituted as hyperplane equations that intersect on the secret point. Secret point is reconstructed from the intersection of at least k hyperplanes.

After these two pioneering research, Asmuth–Bloom and Mignotte proposed (k, n) threshold secret sharing schemes based on the Chinese Remainder Theorem (CRT) in 1983 (Asmuth and Bloom, 1983; Mignotte, 1983). They use special sequences of integers along with the CRT. However, the former technique is more secure than the latter due to the use of random parameters.

Both Blakley’s and Shamir’ secret sharing approaches are used to share digital images among participants recently. Chen and Fu (2008) proposed a new secret image sharing technique using Blakley’s secret sharing scheme based on geometric properties. In another study, Tso (2008) used Blakley’s approach to both share a secret image among participants and enhance the size expansion ratio of shares.

Thien and Lin (2002) used Shamir’s polynomial approach to share a gray level secret image. Intensity values of the secret pixels...
are in 0–255 range. Shamir’s approach selects a prime value enforcing a unique solution during the recovery procedure. Selected prime value also defines a constraint on the range of the pixel values. Thien and Lin’s method used 251, the largest prime value less than 255. This selection enforces truncation of the pixel values larger than 250 before the encoding process which distorts the secret image. However, their method generates noise like images that attracts attention of the malicious users. Lin and Tsai (2004) proposed a method that uses steganography to hide shared values into cover images. Their method also uses parity bits to provide the method authentication ability. Their method embeds one secret pixel into a $2 \times 2$ pixel cover block using LSB embedding technique. Yang et al. (2007) emphasized that using the parity bit as an authentication mechanism is not an appropriate way. Instead, their technique uses hash functions to authenticate stego blocks. Furthermore, their method prevents distortion of the secret image using Galois Fields. Chang et al. (2008) reports two shortcomings in these methods. One is the weak authentication due to a single authentication bit which cannot protect the integrity of the stego images. The other shortcoming is poor visual quality of the stego images. Their method proposes a technique that uses the CRT to improve the authentication ability of the previous methods. Their method also enhances the visual quality of the stego images. Yang and Ciou (2009) shows that Chang et al.’s scheme does not improve the stego image quality and instead degrades. Their work also reports the correct PSNR for Chang et al.’s scheme.

Lin et al. (2009) employs the modulus operator for decreasing the stego image distortion. Their method permits involved participants to restore a lossless secret image and to reconstruct a distortion free cover image. Their work emphasize that if covers are significant images, even slight distortion may be intolerable. Therefore, reconstruction of the cover image after the revealing procedure is an important issue in this research area as reported by them. Lin et al. uses modulus operator defined in Thien and Lin (2003) to hide the shared pixel values in the cover images. Stego images generated by their method have a PSNR of 43 dB for $k = 4$. Their method is also tested on binary cover images which have approximately 38 dB PSNR. Their method also necessitates adjusting the underflow and overflow conditions. Difference between stego and cover pixels is in $[−3,3]$ range for grayscale cover images and in $[−6,6]$ range for binary cover images. Their method has lower PSNR for binary images due to the underflows and overflows.

Lin and Chan (2010) proposed a method that is based on modulus operation to improve the embedding capacity of Lin et al. (2009). Their method generates stego images with 40 dB PSNR for gray level cover images. However, binary or dithered cover image is not considered by their scheme. Their method has lower PSNR with improved embedding capacity. Difference between stego and cover pixels is in $[−6,6]$ range for grayscale cover images. The method represents the secret data in base $m$ notation. Their results also indicate that cover image pixel values within $[255/m] \times m$, 255 cannot be used to embed the shared pixel values.

Method described in Lin et al. (2009) uses modulus operator proposed in Thien and Lin (2003) to embed the shared values into cover images to improve the stego image quality according to traditional LSB embedding scheme. However, the modulus operator causes underflow and overflow for dithered images. On the other hand, embedding strategy in Lin and Chan (2010) based on modulus operator prevents the use of dithered images as cover images. We use Exploiting Modification Direction (EMD) method proposed in Zhang and Wang (2006) with a specially crafted equation to hide the shared values into cover images with less distortion according to LSB embedding and modulus operator. Using EMD during the embedding procedure ensures the visual quality of stego images independent from the intensity range of the cover images (dithered or grayscale) as shown in the experimental results.

The proposed method generates stego images with improved visual quality even for binary cover images. Other works in the literature indicate that visual quality of the stego images depends on the intensity range of the cover images (Lin et al., 2009; Lin and Chan, 2010). Our method generates stego images with enhanced visual quality in either case: 47 and 43 dB PSNR stego images for gray scale and binary cover images respectively. It enhances the stego image quality compared to other methods reported in the literature regardless of the intensity range of the cover images. The difference between stego and cover pixels is in $[−2,2]$ range for gray scale and in $[−4,4]$ range for dithered cover images. Reduced range difference between the cover and stego pixels provides enhanced stego image quality. Lin et al.’s distortion free secret sharing scheme emphasized that even slight distortion may be intolerable if the shared cover images are significant. Our method also recovers cover images after revealing procedure without distortion. Modulus operator is used to recover the original cover pixel values in the revealing procedure.

The rest of the paper is organized as follows. The details of the EMD method, Lee’s method and specially crafted EMD function used by the proposed method are given in Section 2. The proposed scheme is explained in detail in Section 3. Some of the experimental results are summarized and compared with Lin et al.’s method according to PSNR values of the stego images in Section 4. Conclusions are drawn in Section 5.

2. Literature review

The proposed method uses a modified version of the EMD method during the embedding procedure. The details of EMD method, its modified version called by 8-ary method and specially crafted function based on EMD are discussed in this section respectively.

2.1. Exploiting Modification Direction method

Mielikainen (2006) proposed a method that exploits direction of modification to the cover pixels. Mielikainen’s method is immune to the steganographic attacks because it does not exhibit the asymmetric property of the LSB replacement method. Zhang and Wang (2006) proposed a steganographic method called by EMD to transform the secret data into a stream of secret digits in a $(2n+1)$-ary notational system.

Their method assumes that each secret digit in a $(2n+1)$-ary notational system is carried by $n$ cover pixels and is denoted by $(g_1, g_2, \ldots, g_n)$. Only one pixel is incremented or decremented by one in this group. $2n$ possible ways of modification using a group with $n$ pixels exist. One more case is that no pixel is changed. Therefore, $(2n+1)$ modification can be realized with a group of $n$ pixels.

The secret message to be embedded is converted into $(2n+1)$-ary notational system called by secret digits. Each secret digit varies in $[0, 2n]$ range. EMD method uses an embedding function $f$ as weighted sum function evaluated modulo $(2n+1)$. This function is used to calculate a value for each pixel group as in (1).

$$f(g_1, g_2, \ldots, g_n) = (g_1 \times 1 + g_2 \times 2 + \cdots + g_n \times n) \mod (2n+1) \quad (1)$$

Each cover-pixel group is used to represent one digit in the secret digits. If a secret digit to be embedded into corresponding cover pixel block, $(g_1, g_2, \ldots, g_n)$, is equal to the result of $f(g_1, g_2, \ldots, g_n)$, intensity values of the cover pixels do not change. Otherwise, only one pixel of the cover pixel group has to be modified by either incrementing or decrementing the pixel value by one. Thus, distortion of the cover image due to the embedding procedure is not easy to perceive.
Extraction procedure of the EMD is based on the same embedding function. Stego image is partitioned into stego pixel groups with \( n \) pixels. Embedding function \( f \) defined in (1) is applied into stego pixel groups. Calculated value represents the secret digit in \( 2(n+1) \)-ary notational system. Let a stego pixel group be \( (g_1, g_2, \ldots, g_n) \). Then the secret digit is extracted using the following equation:

\[
f(g_1, g_2, \ldots, g_n) = (g_1 \times 1 + g_2 \times 2 + \ldots + g_n \times n) \bmod (2n+1)
\] (2)

For example, let a cover pixel group be \([137\ 139\ 141\ 140]\) and the embedding function yields 3. Assume that secret digit value is 5. The result of the embedding function calculated with new values of the cover pixel block will be equal to the secret digit if the second pixel is incremented by one.

The EMD embedding method provides high quality stego images of approximately 50 dB PSNR. However, some works in the literature try to increase the embedding capacity of the EMD method.

2.2. The 8-ary EMD method

Lee et al. (2007) proposed a method with higher stego image quality and embedding capacity based on EMD. Their method transforms secret messages into the 8-ary notational system, and the cover image is grouped into two consecutive pixels. In their method, one secret digit can be embedded by two cover pixels. Their work revises the EMD embedding function to improve the embedding rate. The number of secret bits embedded per cover pixel is the embedding rate \( R \). A digit in the \((2n+1)\)-ary notational system, representing \( \log_2(2n+1) \) bits, is embedded into \( n \) pixels, among which one pixel is incremented or decremented by 1 with a probability of \( 2n/(2n+1) \). Thus, the embedding rate \( R \) is defined as in (3).

\[
R = \frac{\log_2(2n+1)}{n}
\] (3)

The embedding rate decreases as \( n \) increase. EMD method has maximum embedding capacity when the 5-ary notational system is used. Each secret digit is carried out by two pixels in this case. However, the method has a lower embedding capacity for larger values of \( n \). Lee et al. proposed a method to enhance the embedding rate of the EMD method. The embedding rate of their method is 1.5 times that of the EMD embedding method without losing the stego image quality.

Their method transforms the secret image into a sequence of secret digits in 8-ary notational system. Cover image is grouped into two sequence pixels, \((g_1, g_2)\). A cover pixel group can be used to carry one secret digit in 8-ary notational system. A secret digit is coded into corresponding cover pixel block using function given in (4).

\[
f(g_1, g_2) = (g_1 \times 1 + g_2 \times 3) \bmod 8
\] (4)

Embedding function used by their method is different from the Zhang et al.’s embedding function. Only one pixel is incremented or decremented by one if the value of extraction algorithm is not equal to the secret digit. For example, consider a cover pixel group \([88\ 93]\) with secret digit 5 in 8-ary notational system. Since the result of the embedding function is calculated as 7, the values of \((g_1, g_2)\) are changed to be \((g_1 + 1, g_2 - 1)\) which yields 5.

2.3. Specially crafted EMD function used by the proposed method

EMD method proposed in 2006 uses embedding function, \( f_e = (c_{11} + 2 \times c_{12}) \bmod 5 \), to hide the secret value in \([0–4]\) range into two cover pixels. Only one pixel is changed at a time to give the secret value as a result of embedding function. Alteration interval for the cover pixels is \([-1,1]\). Lee et al. (2007) uses an embedding function as \( f_e = (c_{11} + 3 \times c_{12}) \bmod 8 \) with the same alteration interval, \([-1,1]\), to represent secret values in \([0,7]\) range. This method allows alteration of two cover image pixel values at the same time different from the EMD method.

If alteration interval is changed to be \([-2,2]\) in Lee et al.’s method, embedding function can hide secret digits in \([0,15]\) range into two cover pixels. However, proposed method needs to hide shared values in \([0,16]\) range as described in the next section. Embedding function is changed by the proposed method as in (5) to hide the shared values in \([0,16]\) range into two cover pixels.

\[
f(g_1, g_2) = (g_1 \times 1 + g_2 \times 4) \bmod 17
\] (5)

Actually, proposed embedding function can hide digits in \([0,19]\) range into two cover pixels with alteration interval \([-2,2]\). Proposed function used 17 as modulo value since shared values does not get the values in \([17,19]\) range.

3. Proposed method

The proposed method consists of sharing and retrieving algorithms. The dealer shares the secret image among \( n \) participants using the sharing algorithm for a \((k, n)\) secret sharing scheme. Sharing algorithm also accommodates embedding procedure that hides shared pixel values into cover images. \( n \) stego images generated by the sharing algorithm are distributed among participants. If any \( k \) or more shares are gathered, secret image will be revealed by the retrieving algorithm which also reconstructs cover images from stego images. Therefore, secret image is revealed and the cover image is reconstructed without distortion by the retrieving algorithm. The details of the sharing and retrieving algorithms are introduced in Sections 3.1 and 3.2 respectively.

3.1. Sharing algorithm

Sharing algorithm uses Shamir’s secret sharing approach to divide the secret among \( n \) participants. Each participant gets a share called stego image. Secret is a gray level image where each pixel has intensity value in \([0,255]\) range. The dealer selects a gray level cover image \( C \) for the sharing algorithm. Cover image is used to generate \( n \) different stego images that accommodate shared values.

Assume that secret image \( T \) is a gray level image and the size of the secret image is \( N \times M \). \( T = [t_{uv}]_{u=0\ldots N-1, v=0\ldots M-1} \). Sharing algorithm transforms secret image into a vector \( S = [S_i]_{i=0\ldots 16} \), \( i = 1, 2, \ldots, (N \times M \times \log_{17}(255)) \). Each consecutive pair of the vector \((S_i, S_{i+1})\) is actually a representation of the corresponding secret pixel value in base 17 and calculated as in (6). Lin et al. (2009) transforms secret image into base 7 representation. Each pixel of the secret image is represented by three digits in \([0,7]\) range. Lin and Chan (2010) reports PSNR for different base systems according to their method. Their results indicate that the use of base 7 reduces the distortion of the stego image pixels according to their method. If the proposed method uses base 7 notational system, length of the vector \( S \) be \( N \times M \times 3 \). Increasing the size of \( S \) necessitates bigger cover images to accommodate the shared values. Since Shamir’s polynomial necessitates a prime modulus and 17 is the prime number to represent a secret pixel with two digits, the proposed method chooses base 17. Thus, the length of \( S \) becomes \( N \times M \times 2 \) instead of \( N \times M \times 3 \).

\[
s_i = \frac{t_{uv} - t_{uv} \bmod 17}{17}
\]

\[
s_{i+1} = t_{uv} \bmod 17
\]

\[
i = (2 \times v - 1) + (u - 1) \times M \times 2
\]

In this regard, each pixel of the secret image, \( t_{uv} \) is represented by two corresponding elements of the vector \( S, (S_i, S_{i+1}) \). For example
let two pixels of the secret image be (125, 125), corresponding elements of the vector \( S \) are calculated as \(((7, 6); (12, 11))\).

Sharing algorithm divides the vector \( S \) into sections with \( k - 2 \) elements. Total number of sections is \((N \times M)/(k - 2)\) for a secret image of size \( N \times M \). Each section is used to construct Shamir’s polynomial. First \( k - 2 \) coefficients of the polynomial are obtained from the current section. Last two coefficients of the polynomial accommodate information about the corresponding cover pixel values used during the embedding procedure. Shamir’s polynomial is evaluated for \( n \) unique \( x \) values to calculate shares. The embedding procedure of the sharing algorithm hides the shared values into two cover pixels and obtains stego pixel values. Therefore, the size of the cover image must be greater than \((N \times M \times 4)/(k - 2)\).

Steps to be applied for each section of \( S \) can be summarized as follows. Assume that current section to be shared be the first section. \( k - 2 \) coefficients of the Shamir’s polynomial are determined using the elements of this section \((S_1, S_2, \ldots, S_{k-2})\) as in (7). Since elements of the vector \( S \), or secret values to be shared by the polynomial are in \([0, 16]\), 17 is selected as the prime number for Shamir’s polynomial.

\[
F(x) = (S_1 + S_2x + \cdots + S_{k-2}x^{k-3} + px^{k-2} + px^{k-1}) \mod 17
\]

(7)

Let corresponding cover pixel values be used for hiding the shared pixel value \( F(x_k) \) into corresponding stego image pixels be \((c_{11}, c_{12})\). Values of \( p_1 \) and \( p_2 \) in (7) are calculated using (8)

\[
p_1 = c_{11} \mod 9 \quad p_2 = c_{12} \mod 9
\]

(8)

Last two coefficients of the polynomial, \((p_1, p_2)\), are used to recover the original cover pixels from the stego pixels during the retrieving algorithm. Difference between stego and cover pixel can only be in \([-4, 4]\) range. Let first stego pixel value in \( k \)th share be \( st_{11}^k \). This pixel is calculated during the embedding procedure of the sharing algorithm by translating the original cover pixel \( c_{11} \) with an offset value in \([-4, 4]\) range. Participants know that original value of cover pixel is a number in \([st_{11}^k - 4, st_{11}^k + 4]\) range during the retrieving algorithm. This range has nine consecutive numbers. Each number in this range has unique remainder in modulo 9. The proposed method uses unique remainders in modulo 9 to find the original cover pixel value using stego pixel value as shown in Fig. 1. Subscripts of \( st_{11}^k, st_{12}^k \) and \( c_{11} \) are discarded and used as \( x \) and \( c \) to make the figure understandable.

Sharing algorithm calculates \( n \) pairs of \((x_i, F(x_i))\) where \( x_i \) is the serial number of each participant, \( i = 1, 2, \ldots, k, \ldots, n \) using (7). Shared values \( F(x_1), F(x_2), \ldots, F(x_n) \) will be embedded into corresponding stego image pixels using cover image pixels. Proposed method uses one cover image to obtain \( n \) different stego images, \((st_1^1, st_2^1, \ldots, st_n^1)\). Corresponding pixels at \( k \)th stego image \((st_{11}^k, st_{12}^k)\) for the first section are determined as explained below.

Cover pixel group to be used for creating stego image pixels is \((c_{11}, c_{12})\). Embedding function given in (5) is applied on the current cover pixel group \((c_{11}, c_{12})\) as \(F(c_{11}, c_{12}) = (c_{11} + x \times c_{12}) \mod 17\).

Evaluated value of the embedding function is compared with the current shared value \( F(x_k) \). If they are equal, \( F(x_k) = F(c_{11}, c_{12}) \), there is no need to modify cover image pixels. Thus, corresponding pixel values at \( k \)th stego image are equal to the cover image pixel values, \((st_{11}^k, st_{12}^k) = (c_{11}, c_{12})\). Otherwise, cover image pixel values are modified to make \( F(x_k) \) equal to \( F(c_{11}, c_{12}) \). Embedding procedure determines alteration intervals, \((x_1, x_2)\) and \((y_1, y_2)\), for each cover image pixels \((c_{11}, c_{12})\) separately.

Embedding procedure of the sharing algorithm explained below is used to calculate the corresponding pixel values at \( k \)th stego image \((st_{11}^k, st_{12}^k)\) as a function of shared value \( F(x_k) \), cover pixel group \((c_{11}, c_{12})\) and the result of the embedding function \(F(c_{11}, c_{12})\). Steps of the algorithm to determine the values of \( n \) stego image pixels \((st_{11}^k, st_{12}^k)\) for the first section of \( S \) is given below.

3.1.1. Embedding procedure

1. Evaluate the embedding function for current cover pixel group using (5).
2. Determine the alteration intervals for each cover pixel using (9) according to their intensity values.

\[
\begin{align*}
(c_{11} \geq 254) &\Rightarrow x_1 = -4 \quad x_2 = 0 \\
(c_{12} \geq 254) &\Rightarrow y_1 = -4 \quad y_2 = 0 \\
(c_{11} \leq 1) &\Rightarrow x_1 = 0 \quad x_2 = 4 \\
(c_{12} \leq 1) &\Rightarrow y_1 = 0 \quad y_2 = 4
\end{align*}
\]

(9)

3. Repeat the following steps for \( k = 1, \ldots, n \).
4. Check whether \( f_e(c_{11}, c_{12}) = F(x_k) \).

4.1. If equal, no modifications on cover image pixels need to be done. Pixel values of the current stego image are determined to be \((st_{11}^{k1}, st_{12}^{k1}) = (c_{11}, c_{12})\).

4.2. Otherwise follow the steps below.

4.2.1. Repeat the following steps for \( X = x_1 \) to \( x_2 \)
4.2.2. Repeat the following steps for \( Y = y_1 \) to \( y_2 \)
4.2.3. Calculate the value of \( b \) as \( b = (X + 4 \times Y) \)
4.2.4. Check whether \( b + f_e(c_{11}, c_{12})) \mod 17 = F(x_k) \). If they are equal, determine the value of pixels in the current stego image to be \((st_{11}^{k1}, st_{12}^{k1}) = (c_{11} + X, c_{12} + Y)\), and go to the step 2.

Note that, each stego image, \((st_1, st_2, \ldots, st_n)\), is meaningful and similar to cover image \( C \). Sharing procedure controls the overflow and underflow conditions during the embedding procedure. Alteration intervals are also adapted according to current values of cover image pixels. Maximum absolute difference among a stego image pixel and corresponding cover image pixel can be four. Maximum difference can arise when the intensity value of the cover pixel is close to overflow or underflow limit \((c_{12} \leq 2 \times c_{11} \geq 254)\).

Two examples will be given below. Both examples use (4, 5) secret sharing scheme. Let the secret value be \( 202 \). Cover pixels used are assumed to be \((193, 199)\) for the first and \((253, 255)\) for the second example. Second example tests a cover image pixel close to the overflow limit, \(255\). Sequential number of each participant is selected as \(x_k = k, \quad k = 1, 2, \ldots, 5\).

Example 1. Secret value 202 is transformed into a vector \( \{11, 15\} \) with two elements. Vector \( S \) is divided into sections with \( k - 2 = 2 \) elements. One section is formed to be \((11, 15)\) due to the size of the vector. Remainders of cover pixels in modulo 9 are calculated by using (8) as \( p_1 = 4 \) and \( p_2 = 1 \). Shamir’s polynomial is then constructed as \( F(s) = (11 + 15x + 4x^2 + x^3) \mod 17 \). Shared values for each participant are then calculated by using their own serial numbers as \(14, 14, 0, 12\) and \(5\). We show the details of the embedding section of the sharing procedure for the first shared value, \(14\), on two different examples with distinct cover pixel groups.

First, the embedding function \(f_e\) is evaluated for the current cover pixel group as \( f_e = (193 + 4 \times 199) \mod 17 = 3 \) which indicates that the stego image pixels should be modified since the value of \( f_e \) is not equal to the shared value \( 14 \). The following procedure is the implementation details of Step 5.

Alteration range for both cover pixels are \([-2, 2]\) since cover pixels are not close to the limits. The proposed method evaluates \(f_e\) to be equal to the shared value if first and second cover pixels are decremented by two and one respectively at the 4.2 step of the
embedding procedure. Therefore, corresponding stego pixels in the first stego image become (191, 198). The shared value, 14 is revealed by evaluating \( f_e \) on the stego image pixels since \( f_e = (191 + 4 \times 198) \mod 17 = 14 \). Stego pixels in other four stego images become (191, 198), (194, 198), (193, 197) and (191, 200) respectively. Maximum absolute difference among the cover and stego pixels cannot exceed two since cover pixel values are not close to the limit.

**Example 2.** Since both \( p_1 \) and \( p_2 \) are 3 \((255 \mod 9)\), Shamir's polynomial is constructed as \( F(x) = (11 + 15x + 3x^2 + 3x^3) \mod 17 \).

Shared values for each participant are then calculated by using their own serial numbers as 15, 9, 11, 5 and 9.

Evolution of \( f_e \) yields 0 for current cover pixel group, \( f_e = (255 + 4 \times 255) \mod 17 = 0 \). The alteration range for both cover pixels are \([-40]\) since it is not possible to increment the values without causing overflow. Embedding function evaluated using altered pixel values results in the first shared value 15, if only the first cover pixel is decremented by two. Therefore, corresponding stego pixels in the first stego image are determined as \((253, 253)\) which evaluates the shared value \( f_e = (253 + 4 \times 255) \mod 17 = 15 \). Stego pixels in other four stego images become \((253, 253)\), \((253, 254)\), \((255, 252)\) and \((255, 253)\). Maximum absolute difference among the cover and stego pixels is 3 for the third stego image \((255, 252)\). Cover pixel values close to the overflow limit cause a greater alteration magnitude.

### 3.2. Retrieving algorithm

The algorithm requires at least \( k \) stego images from the participants to successfully retrieve the secret image. Cover image used during the sharing procedure is also reconstructed in this stage. Stego images gathered from participants are gray level images. Retrieving algorithm partitions stego images into two pixel sections. Embedding function used in sharing procedure, \( f_e \), is applied on each section to reveal the shared values, \( F(x_1), F(x_2), \ldots, F(x_k) \).

Lagrange’s interpolation technique is used to interpolate the polynomial from \( k \) distinct points as suggested by Shamir. First \( k - 2 \) coefficients of the polynomial yields the corresponding secret pixel values in base 17 representation whereas the last two coefficients of the polynomial are used by the retrieving algorithm to reconstruct corresponding cover image pixels. Retrieving algorithm contains a procedure called reconstruction. The reconstruction procedure uses a two pixel group from any stego image and last two coefficients of the Shamir’s polynomial to retrieve cover pixel values.

Assume that the dealer gathers \( k \) stego images from any \( k \) participants, \((s_1^1, s_2^1, \ldots, s_k^1)\). Each stego image is divided into sections of two consecutive pixels. Embedding function \( f_e \) is applied on each section of the stego image. The sharing algorithm necessitates stego images of size at least \( ((4NM)/(k - 2)) \) for a secret image of size \( NM \). Therefore, the minimum size of the stego image can be \( 2NM \) when \( k = 4 \) for the proposed method. Corresponding sections of \( k \) stego images are used to obtain shared values as in \((10)\).

\[
F(x_k) = (s_{g+4}^k + 4 \times s_{g+1}^k) \mod 17 \quad (10)
\]

Shared values are used to interpolate Shamir’s polynomial using Lagrange’s interpolation technique. Lagrange’s interpolation technique is applied on \( k \) different points \(((x_1, F(x_1)), (x_2, F(x_2)), \ldots, (x_k, F(x_k))\) and the following polynomial is obtained.

\[
F(x) = (s_1 + s_2 + \cdots + s_{k-2} + p_1 + p_2) \mod 17 \quad (11)
\]

First \( k - 2 \) coefficients of the polynomial constitute the corresponding elements of the vector \( S \) whereas the last two coefficients are used to reconstruct the values of corresponding cover pixels. The following procedure is used to reconstruct cover image pixel values using last two coefficients, \((p_1, p_2)\).

#### 3.2.1. Reconstruction procedure

A section from any \( k \) stego images is used in this procedure. Assume that \( k \)th stego image is used to construct the original values of cover pixels. In this case, section to be used is \((s_{t_1}^{t_1}, s_{t_2}^{t_1})\) from \( k \)th stego image. Each cover pixel \( c_{11} \) and \( c_{12} \) are reconstructed using corresponding stego pixels \( s_{t_1}^{t_1} \) and \( s_{t_2}^{t_1} \) respectively. The algorithm to reconstruct corresponding cover pixel \( c_{11} \) is explained on the first pixel \( s_{t_1}^{t_1} \) in the current section for demonstration purposes and can be applied to reconstruct the second cover pixel using \( s_{t_2}^{t_1} \). Values of \((p_1, p_2)\) used in the following procedure have been obtained in \((11)\).

1. Determine the control range, \([b_1, b_2]\), using the current stego pixel.

\[
(s_{t_1}^{t_1} \leq 3) \Rightarrow b_1 = 0, \quad b_2 = s_{t_1}^{t_1} + 4
\]

\[
(s_{t_1}^{t_1} \geq 252) \Rightarrow b_1 = s_{t_1}^{t_1} - 4, \quad b_2 = 255
\]

\[
(4 < s_{t_1}^{t_1} < 252) \Rightarrow b_1 = s_{t_1}^{t_1} - 4, \quad b_2 = s_{t_1}^{t_1} + 4
\]

2. Repeat the following steps for \( b = b_1, \ldots, b_2 \).

2.1. Check if \( b \mod 9 = p_1 \).

2.1.1. If true, corresponding pixel in reconstructed cover image \( c_{11} \) is determined to be \( b \).

2.1.1. Otherwise, continue to loop.

Reconstruction procedure controls all the values in \([b_1, b_2]\) range to find a value that has remainder \( p_1 \) in modulo 9. Steps listed above are also applied on \( s_{t_1}^{t_2} \) to determine the value of \( c_{12} \) using \( p_2 \). As a result, corresponding original cover pixel values for current stego blocks are reconstructed.

Retrieving algorithm is applied on each section of the stego images. Vector \( S \) is constituted to accommodate secret pixel values in base 17 notation. \( S \) is partitioned into groups of two elements. Each group is the base 17 representation of the corresponding secret pixel. Assume that first two elements of \( S \) be \((s_1, s_2)\). In this case, first pixel of the secret image is calculated by \((s_1 \times 17 + s_2)\). Thus, the retrieving algorithm both reveals the secret image and reconstructs the original cover image from any \( k \) stego images.
The steps of the retrieving algorithm are given below:

1. Dealer gathers any $k$ stego images from $k$ participants, $(st_1, st_2, \ldots, st_k)$.
2. Each stego image is partitioned into sections of two consecutive pixels.
3. Repeat the following steps for corresponding sections of the stego images.
   3.1 Shared values $F(x_1), F(x_2), \ldots, F(x_k)$ are obtained from the corresponding sections of the $k$ stego images using (10).
   3.2 Lagrange’s interpolation technique is used to interpolate the polynomial for the current shared values as in (11).
3.3 First \( k - 2 \) coefficients of the polynomial constitute the corresponding elements of the vector \( S \). Last two coefficients of the polynomial and a section from any stego image are used to reconstruct corresponding cover image pixels as defined in the reconstruction procedure.

4 Vector \( S \) constituted in Step 3 is partitioned into sections of two elements. Each section is used to retrieve the corresponding secret pixel value by \( s_i \times 17 + s_i,1 \).

The first example given above in the sharing procedure is also used in this section to demonstrate the implementation details of the algorithm. A \((4,5)\) secret sharing scheme is assumed.

Stego pixel values are calculated as \((191,198), (191,198), (194,198), (193,197)\) and \((191,200)\) for the first example. Any four shares are gathered by the dealer during the retrieving procedures.

**Example 3.** Assume that the dealer selects \((191,198), (194,198), (193,197)\) and \((191,200)\) stego images among the five shares. Each stego image is partitioned into sections of two consecutive pixels. Shared values are obtained using \((10)\).

Shared value for the second stego image is calculated using embedding function, \( F(x_2) = f'_2(191,198) = (191 + 4 \times 198) \mod 17 = 14 \). Other shared values are obtained in the same way as \( F(x_3) = 0, F(x_4) = 12, F(x_5) = 5 \). Lagrange's interpolation technique is applied on these values to reconstruct Shamir's polynomial as \( F(x) = (11 + 15x + 4x^2 + 1x^3) \mod 17 \). First \( k - 2 \) coefficients constitute the corresponding elements in the vector \( S \) whereas the last two coefficients, \((p_1, p_2)\), are used to reconstruct original values of corresponding cover pixels. A pixel group from any stego image is sufficient during the reconstruction of the cover pixels. Assume that the first group \((191,198)\) is used and the control range for the first stego pixel is determined as \[187,195]\). Control range for the second stego pixel is determined likewise as \[194,202]\). First and second control range are used to reconstruct corresponding consecutive two original cover pixel values.

Numbers in the first control range have corresponding remainders \((7, 8, 0, 1, 2, 3, 4, 5, 6)\) in modulo 9 and \( p_1 \) is calculated as 4 from Shamir's polynomial. Number in the first range whose remainder is equal to \( p_1 \) is the corresponding cover pixel value. In this regard, 193 is 4 in modulo 9 and is the original pixel value of the corresponding cover pixel. Second control range is used to reconstruct the second stego pixel value. Numbers in the second control range have the corresponding remainders \((5, 6, 7, 8, 0, 1, 2, 3, 4)\) in modulo 9 and \( p_2 \) is calculated as 1. Only one of the numbers in this control range, 199 is 1 in modulo 9 and is the original value of corresponding cover pixel. Therefore corresponding cover pixel values are determined to be \((193,199)\).

Vector \( S \) is partitioned into sections of two elements. Only one section is created in this example due to the size of the vector, \((11,15)\). Each section reveals the corresponding secret value. Secret value is calculated as \((11 \times 17 + 15) = 202\).

**4. Experimental results and discussion**

In this section, we give the experimental results and discuss the efficiency, accuracy of the proposed method.

**4.1. Experimental results**

This section demonstrates the performance of the proposed method using grayscale and dithered cover images. Test images used as cover images during the experiments are given in Fig. 2. Fifteen different grayscale images and their dithered versions of size \(512 \times 512\) pixels are selected to demonstrate the effectiveness of the proposed method on different images with varying intensity values. Fig. 3 shows the gray level secret image of size \(256 \times 256\) to be shared among the participants. Visual quality of the stego images generated by the proposed method is quantitatively determined using Peak to Signal Noise Ratio (PSNR). Higher PSNR indicate better quality or more natural looking stego image. PSNR is defined as in \((12)\).

\[
\text{PSNR} = 10 \log_{10} \frac{I^2}{\text{MSE}} \text{dB}
\]

where MSE is the mean square error between the cover image and stego image and \( I \) is the maximum intensity value used in images. MSE for a cover image of \(2N \times 2M\) is

\[
\text{MSE} = \frac{1}{2M \times 2N} \sum_{i=1}^{2N} \sum_{j=1}^{2M} (c_{ij} - st_{ij})^2
\]

where \( c_{ij} \) and \( st_{ij} \) are the pixel values of cover and \( k \)th stego images respectively. As Eqs. \((12)\) and \((13)\) indicate, PSNR increases as MSE or the sum of squared differences between the cover and stego image decrease.

Eight bit depth gray level cover images are used for the first experiment. A \((4,4)\) secret sharing scheme is used to test the visual quality of the generated stego images. Secret image in Fig. 3 is partitioned among four participants using the proposed method. Each share image is the modified version of the corresponding cover image and called a stego image. Stego images and corresponding PSNR values are given in Fig. 4(a)-(d). Stego images generated have approximately 48 dB PSNR on the average as can be seen in figure. Fig. 4(e) shows the secret image revealed during the retrieving algorithm. Secret image is revealed if all four of the stego images are gathered and has infinite PSNR indicating lossless characteristic of the method. Original cover image is also reconstructed during the revealing algorithm using any one of the stego images. Reconstructed cover image with infinite PSNR is given in Fig. 4(f).

Proposed method uses only \((256 \times 256 \times 4/2) = 256 \times 512\) pixels of the cover image during the sharing algorithm for \(k = 4\) and secret image size of \(256 \times 256\). Rest of the pixels of the cover image, \((512 \times 512 - 256 \times 512) = 131,072\), are not used by the algorithm for coding purposes. The relation between the number of secret pixels and the number of cover pixels is given in \((14)\).

\[
\frac{N_T \times M_T \times 4}{(k - 2)} \leq N_C \times M_C
\]

![Fig. 3. Gray level secret image of size 256 × 256.](image)
Fig. 4. Four shares, revealed secret and recovered cover for a (4,4) scheme.
Fig. 5. Four shares, revealed secret and recovered cover for a \(\text{A}(3,4)\) scheme \((r = 1)\).
Fig. 6. Original cover, three shares, revealed secret and recovered cover for a (3,3) scheme with dithered cover ($ur = 1$).
Let the total number of cover pixels used by the sharing algorithm be denoted by UCP and total number of pixels of the cover image denoted by CP. The proposed method make use of a metric called usage ratio \( ur \) given in (15) to evaluate the PSNR of stego images. For the first experiment, a \( ur \) value of 0.5 is selected.

\[
ur = \frac{UCP}{CP}, \quad ur \in [0, 1]
\]  

(15)

Maximum height or width of a secret image with aspect ratio 1 is defined by the cover image size \( N_c \times M_c \), \( k \) and \( ur \) as in (16).

\[
N_T = M_T = \sqrt{N_c \times M_c \times (k - 2) \times ur} \frac{4}{4}
\]  

(16)

The value of \( k \) is reduced to 3 for the second experiment in order to use all cover pixels for coding with \( 512 \times 512 \) cover image. Width and height of the secret image is determined as \( \sqrt{\frac{512 \times 512 \times (3 - 2) \times 1}{4}} = 256 \) using (16). A secret image of size \( 256 \times 256 \) pixels given in Fig. 3 is selected as the secret image. Four stego images and corresponding PSNR values generated by the second experiment are given in Fig. 5(a)–(d). Stego images exhibit approximately 45.9 dB PSNR on the average even if all pixels of cover image are used for coding. Revealed secret and reconstructed cover images of infinite PSNR are given in Fig. 5(e) and (f) respectively. The proposed method generates stego images with approximately 45 dB PSNR even if it uses all pixels of the cover image for coding. Both experiments indicate that the proposed method generates stego images with improved visual quality.

A dithered binary image of a gray level image used in the first two experiments is selected as the cover image for the third experiment as given in Fig. 6(a). The secret image in Fig. 3 is encoded in a \( 512 \times 512 \) pixel binary cover image shown in Fig. 6(a) with (3, 3) secret sharing scheme for \( ur = 1 \). Generated stego images and corresponding PSNR values are given in Fig. 6(b)–(d). The revealed secret and reconstructed cover images with infinite PSNR are given in Fig. 6(e) and (f) respectively. Stego images have approximately 41 dB PSNR on the average. The difference between generated stego images and corresponding cover pixels are in [−4, 4] which makes stego images indistinguishable from covers by the human visual system.

Another experiment is realized to evaluate the quality of the proposed method in terms of the human visual system using wPSNR (weighted Peak Signal to Noise Ratio) (Makoto, 1998). Table 1 lists the qualities of the stego images with test images given in Fig. 2 using the (4,4) threshold scheme with secret image of size \( 313 \times 313 \). The average PSNR of four stego images is approximately 46.79 dB and 42.9 dB for gray level and dithered test images respectively. The high values of wPSNR (59.68 dB and 57.75 dB for gray level and dithered test images) indicate that the fidelity of stego images is high enough after the embedding procedure.

Fig. 7. PSNR of the three methods with both gray and binary covers for \( ur = 0.75 \).

Fig. 8. PSNR of the three methods for different \( ur \) values with gray level cover images.

Experiments reported above indicate the accuracy of the proposed method that also exhibits superiorities compared to similar works. In this regard, more experiments are realized to show the performance of the method on the visual quality of stego images. PSNR and \( ur \) are used as metrics to show the capabilities of the method.

In order to make a fair comparison of the proposed and other methods, an average \( ur \) value of 0.75 is selected for all methods compared in the first experiment. Thus, the embedding efficiency of all methods can be compared. The \( ur \) value of the proposed method becomes 0.75 for \( k = 4 \) and a \( 313 \times 313 \) secret and a \( 512 \times 512 \) cover image. Width or height of the secret image is determined by (16), \( \sqrt{\frac{512 \times 512 \times (3 - 2) \times 0.75}{4}} = 313 \). A gray cover image (Lena) and its dithered version of size \( 512 \times 512 \) pixels are used to demonstrate the effectiveness of all methods. The PSNR of stego images of three methods (proposed method and two related works) are evaluated for these cover images. Fig. 7 illustrates the PSNR values of stego images of three works for \( ur = 0.75 \). Proposed method has higher PSNR with respect to other works reported in the literature for the same \( ur \). Proposed method has approximately 43 dB PSNR even if cover is a dithered image. Lin et al. (2009) generates stego images with 38 dB PSNR when the cover image has two levels of intensity. Lin and Chan (2010) does not support binary cover images at all. Proposed method has also higher PSNR for gray level cover images as shown in Fig. 7.

PSNR of stego images for some \( ur \) values are measured in the second experiment. Line graphs in Fig. 8 indicate the PSNR of three works for \( ur \) values from 0.25 to 1.0. Gray level cover image of size \( 512 \times 512 \) given in Fig. 3 is used to test all three methods. The secret image is resized such that all three methods have equal \( ur \) values. As expected, the visual quality of stego images decrease as the value of \( ur \) is increased. The proposed method yields higher PSNR for all \( ur \) values compared to other studies as shown in the figure. All pixels of the cover image have been used during the embedding procedure for \( ur \) value of 1. Proposed method generates stego images with approximately 45 dB PSNR even if all pixels of the cover have been used for coding purposes.

The secret image is resized for all methods for equal \( ur \) while comparing PSNR values as mentioned above. In this experiment, size of the cover image is selected to be \( 2N \times 2M \) for a secret image of size \( N \times M \) as in the literature to investigate the effect of threshold (k) value on the \( ur \) and PSNR values. Proposed method (Lin et al.,...
Table 1
PSNR and wPSNR values of the proposed method.

<table>
<thead>
<tr>
<th>Test images</th>
<th>Gray level test images</th>
<th>Dithered test images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average wPSNR of four stego images (dB)</td>
<td>Average PSNR of four stego images (dB)</td>
</tr>
<tr>
<td></td>
<td>Average PSNR of four stego images (dB)</td>
<td>Average PSNR of four stego images (dB)</td>
</tr>
<tr>
<td>Airplane</td>
<td>59.65</td>
<td>56.83</td>
</tr>
<tr>
<td>Cameraman</td>
<td>59.67</td>
<td>55.39</td>
</tr>
<tr>
<td>Clown</td>
<td>59.50</td>
<td>55.36</td>
</tr>
<tr>
<td>Couple</td>
<td>59.77</td>
<td>60.06</td>
</tr>
<tr>
<td>Elaine</td>
<td>59.69</td>
<td>58.89</td>
</tr>
<tr>
<td>Goldhill</td>
<td>59.70</td>
<td>58.58</td>
</tr>
<tr>
<td>House</td>
<td>59.81</td>
<td>58.26</td>
</tr>
<tr>
<td>Lena</td>
<td>59.67</td>
<td>59.00</td>
</tr>
<tr>
<td>Mandrill</td>
<td>59.65</td>
<td>59.91</td>
</tr>
<tr>
<td>Peppers</td>
<td>59.70</td>
<td>58.42</td>
</tr>
<tr>
<td>Sailboat</td>
<td>59.72</td>
<td>56.97</td>
</tr>
<tr>
<td>Splash</td>
<td>59.76</td>
<td>57.39</td>
</tr>
<tr>
<td>Tiffany</td>
<td>59.60</td>
<td>54.82</td>
</tr>
<tr>
<td>Einstein</td>
<td>59.60</td>
<td>59.52</td>
</tr>
<tr>
<td>Zelda</td>
<td>59.66</td>
<td>56.85</td>
</tr>
<tr>
<td>Average</td>
<td>59.68</td>
<td>56.80</td>
</tr>
</tbody>
</table>

Fig. 9. Usage ratios of the three methods for different threshold values.

2009; Lin and Chan, 2010) have ur values 1/(k − 2), 3/(4k × (k − 3)) and 3/(4k × (k − 1)) respectively. In other words, the value of ur is a function of the threshold value k for all methods. Usage ratios of all three methods are plotted for k values from 3 to 9 in Fig. 9. As the threshold value is increased, the number of cover pixels used decreases for all methods which also improve the PSNR values of stego images. Also, difference among the ur values of three works decreases as the threshold is increased. So, PSNR of stego images for different threshold values are measured for three methods to give an idea about the embedding capability of the methods. Fig. 10 shows the relationship between the threshold value and PSNR of stego images. Proposed method generates stego images with higher PSNR than that of other methods for different threshold values. For example, for k = 6 the proposed method’s ur value is approximately 0.25 as can be seen in Fig. 9, Lin et al. (2009) and Lin and Chan (2010) have approximately 0.25 and 0.15 ur values respectively. From Fig. 10, the proposed method Exhibits 4 dB more PSNR compared to other methods for k = 6. The proposed method also generates higher PSNR (approximately 49 dB) compared to others at the threshold value 4, which yields a maximum difference among the ur values.

Third experiment uses difference image D defined in (17). Absolute difference among the cover image and stego image pixel values determines the pixel values of difference image.

\[ D = |d_{ij} - d_{ji}| \]

Fig. 10. PSNR of the three methods for different threshold values.

Table 2
Intensity range of difference images for three methods with gray level and dithered covers.

<table>
<thead>
<tr>
<th>Intensity range of difference image</th>
<th>Gray level cover image</th>
<th>Dithered cover image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method</td>
<td>[0,2]</td>
<td>[0,4]</td>
</tr>
<tr>
<td>Lin et al. (2009)</td>
<td>[0,3]</td>
<td>[0,6]</td>
</tr>
<tr>
<td>Lin and Chan (2010)</td>
<td>[0,6]</td>
<td>Does not support</td>
</tr>
</tbody>
</table>

Difference image gives an idea about the alteration existed on cover images. Intensity range of difference image is related with used sharing scheme and intensity range of the cover image. Proposed method and other works in the literature generate difference images with different intensity range. When difference image has narrow intensity range, stego image has better visual quality. Otherwise, visual quality of the stego image deteriorates. Table 2 lists the intensity range of difference images for three methods when two different cover images (gray level and its dithered version) are used. Proposed method generates difference image with [0,2] intensity range for gray level cover images. Difference images for other works have broader intensity ranges (Lin et al., 2009; Lin and Chan, 2010). In other words, their absolute difference among the cover and stego images are greater than that of the proposed method for gray level cover images. Also, the proposed method generates difference image in [−4,4] range for dithered cover images. Lin et al.’s (2009) method has a difference image in [−6,6] range for dithered cover images which deteriorates their PSNR values. Lin and Chan’s (2010) work does not support binary cover images due to the overflow effect. Thus, the proposed method exhibits higher PSNR for stego images with respect to other works for both gray level and binary cover images.
Table 3 compares the proposed method with related methods that ensure recoverable cover image. Proposed method has a higher PSNR for gray level and dithered cover images. Proposed method increases PSNR approximately 4 dB and 7 dB for gray level cover images compared to other works respectively. Lin and Chan (2010) does not support two level cover images. Lin et al. (2009) generates stego images with approximately 38 dB PSNR when the cover image has two intensity levels. However, the proposed method generates stego images with 43 dB PSNR for the dithered cover image. The proposed method increases PSNR approximately 5 dB for two intensity value cover images. Embedding capacity of the proposed method’s is also higher if improvement on stego images’ visual quality is taken into account.

4.2. Discussion and security analysis

Efficiency and accuracy of the proposed method is discussed in this section. Two of the metrics, PSNR and the secret capacity are used to compare the method with others reported in the literature. There is a tradeoff between PSNR and secret capacity and proposed method aims to improve PSNR while preserving the secret capacity.

Similar works are affected from the intensity range of the cover images. These methods use modulus operator to hide the shared values. Lin et al. (2009) utilize modulus operator to embed the shared values to diminish the distortion of the cover images. They used 7 as the prime value and stego pixels are incremented or decremented by 7 to recover the shared value. This modification decreases stego image quality when the cover image is dithered or a gray level image with many black and white pixels. Lin and Chan (2010) also utilize the modulus function to hide the shared values. They named cover pixels in [252, 255] range as excess pixels and did not use those to avoid overflow during embedding. PSNR is not reported for dithered binary cover images since white pixels cannot be used for hiding shared values.

Both of these methods are affected by overflow or underflow due to the strategy used by the embedding procedure. A modified EMD function to generate less distorted stego image for even binary cover images is proposed in this work. Modified embedding function can code shared values in [0, 16] range on adjacent cover pixels. Table 4 lists modifications on cover pixels for shared values in [0, 16] range to recover them back. The accuracy of the embedding function can be verified if four scenarios listed below are considered.

- Scenario 1: Adjacent cover pixels have intensity values in [2, 253] range.
- Scenario 2: Adjacent cover pixels are in [0, 1] range.
- Scenario 3: Adjacent cover pixels are in [254, 255] range.
- Scenario 4: First cover pixel is in [0, 1] range and second cover pixel is in [254, 255] range.

Shared values can be coded into cover pixels with extreme intensities by modifying them in [−4, 4] range. A modification range of [−2, 2] is appropriate for cover pixels in [2, 253] range which improves PSNR for natural images. Table 4 indicates that the proposed method can hide shared values into corresponding cover pixels depending on intensity range of cover pixels and the difference \( d \) between embedding function \( f_e \) and current shared values in base 17.

In order to verify the table, examples will be given from two different scenarios. For the first scenario, adjacent cover pixels’ intensity values are assumed to be 142 and 156. In this case embedding function in (5) yields 1. Let current shared value be 13. Difference \( d \) between the shared value and \( f_e \) is calculated to be 12 at base 17. The value at the column for \( d = 12 \) at the first row \((-1, -1)\) should be used to modify the cover pixels. New intensity values of stego pixels become 141 and 155 respectively. After the modification, embedding function yields the shared value, \((141 + 155 \times 4) \mod 17 = 13\).

For the third scenario, adjacent cover pixel values are assumed to be 254 and 255 for a shared value of 8. Embedding function for cover pixels yields \((254 + 255 \times 4) \mod 17 = 16\). Difference between the shared value and \( f_e \) is \(-8 = 9 \mod 17\). The value at the column for \( d = 9 \) at the third row \((0, -2)\) should be used to modify the cover pixels. Stego pixels become 254 and 253 after the modification which yields the shared value \((254 + 253 \times 4) \mod 17 = 8\) by the EMD function during the retrieving algorithm.

Other methods can cause more distortion for cover pixels with extreme intensities. Gray level and dithered cover images of 256 x 256 are used to demonstrate the performance of the hiding algorithm for both proposed and Lin et al.’s (2009) methods for \( ur = 1 \). Histograms of difference between cover and stego images are computed and plotted in Fig. 11 to show the performance of the hiding algorithm.

Histograms of difference images of both methods are given in Fig. 11(a) and (b) for gray level cover image. Proposed method did not modify 21% of the cover pixels whereas Lin et al.’s (2009) method could achieve 14% for the same cover. Besides, the proposed method generates less modified stego image indicated by a narrow histogram compared to other method’s.

Difference image histograms for dithered cover image are given in Fig. 11(c) and (d). Proposed method modifies the cover less than Lin et al.’s (2009) method indicated by a narrow histogram. Difference image histograms clearly demonstrate that the proposed method distorts cover images less than similar methods and generates stego images with higher PSNR even if dithered cover images are used.

The reason to choose 17 as the prime value by the sharing algorithm will be explained in this section. The method aims to hide shared values into cover images using modified EMD method rather than LSB embedding or modulus operator to improve the PSNR of stego images regardless of cover image intensity range. Similar works use base 7 to represent the shared values. The number of shared values would be \((3NM) / (k-2)\) if proposed method would have used base 7 like others. First \( k-2 \) coefficients of the Shamir’s polynomial are constituted from corresponding shared digits. Last two coefficients are used to make the method reversible. Since each shared value is coded into corresponding two cover pixels, size of the cover image must be at least \((6NM) / (k-2)\). Thus secret capacity of the proposed method would be lower than similar works. The size of the cover image must be at least \((4NM) / (k-2)\), if 17 is selected as modulus for Shamir’s polynomial. Prime modulus 17 make the method’s secret capacity comparable while it ensures the

<table>
<thead>
<tr>
<th>(k, n) threshold</th>
<th>Meaningful share image</th>
<th>Quality of gray level share images (dB)</th>
<th>Quality of binary share images (dB)</th>
<th>Lossless secret image</th>
<th>Extra expansion</th>
<th>Lossless cover image</th>
<th>Maximum capacity (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin et al. (2009)</td>
<td>Yes</td>
<td>Yes</td>
<td>43</td>
<td>38</td>
<td>Yes</td>
<td>No</td>
<td>Yes (−17)N−M</td>
</tr>
<tr>
<td>Lin and Chan (2010)</td>
<td>Yes</td>
<td>Yes</td>
<td>40</td>
<td>–</td>
<td>Yes</td>
<td>No</td>
<td>Yes (−17)N−M</td>
</tr>
<tr>
<td>Proposed method</td>
<td>Yes</td>
<td>Yes</td>
<td>47</td>
<td>43</td>
<td>Yes</td>
<td>No</td>
<td>Yes (−25)N−M</td>
</tr>
</tbody>
</table>
improved PSNR of stego images. Fig. 12 illustrates the secret capacity of the proposed method and similar works for a 512 × 512 cover image. Proposed method’s capacity is higher than Lin et al.’s (2009) for k ≤ 6. Even though Lin and Chan’s (2010) method has higher secret capacity, it cannot use dithered images as cover and has a lower PSNR than the proposed method for gray level covers.

Reversible property of the proposed method using modulus operator will also be discussed in this section. Let current cover pixel be c and t denotes c in modulo 9. Embedding procedure modifies the cover pixel to hide the shared value. Worst case modification range is [−4,4] and corresponding stego pixel values is in [c−4, c+4] range. The numbers in this range are [t−4, t−3, . . . , t, . . . , t+3, t+4] in modulo 9 respectively. The value that yields t in modulo 9 is unique and this value corresponds to the original value of the cover pixel. Accommodating the cover pixel in modulo 9 is adequate since modification interval is 9. Thus, last two coefficients of Shamir’s polynomial are used to reconstruct the corresponding cover pixel correctly.

Secret can be reconstructed by processing at least k shares from participants. Lagrange interpolation does not reveal information about secret unless k shares are available. Coefficients of a k – 1 degree polynomial can be computed even if a single share out of k is missing. Those coefficients do not reveal secret at all. One can try to use Lagrange’s interpolation technique to reconstruct a kth degree polynomial from k – 1 points but secret cannot be revealed unless the kth point is guessed. There exist m unique solutions since Shamir’s polynomial is in Zm. Let the event of hitting the (k – 2) secret digits be Ej for j = 1, 2, . . . , (NM⌈log255⌉/(k − 2)). Thus, we have Pr[Ej] = 1/m. The probability Pr[S] of obtaining secret from the cooperation of k−1 participants is 17−⌈log255⌉(MN)/(k−2).
5. Conclusion and future work

An active research topic in the secret image sharing field is the recovery of the cover image from stego images during the revealing procedure recently. If the shared cover images are significant, even slight distortion may be intolerable. Lin et al. (2009) uses cover images during the sharing procedure regardless of their intensity range. However, visual quality of the stego images depends on the intensity range. The method generates stego images with 43 dB PSNR for gray level but deteriorates to 38 dB for binary cover images due to the use of modulus operation to hide shared pixel values. Lin and Chan (2010) aims to enhance the embedding capacity of this method. However their method has some drawbacks. Binary cover images are not supported by their method. PSNR of stego images are reported as 40 dB for m = 7 as suggested in their work. Besides, cover pixels in \([255/m \times m, 255]\) range are not used by their method. Stego image visual quality improvement regardless of the intensity range of cover images is aimed in the proposed method. The method uses EMD technique and modulus operator during the sharing procedure. PSNR of stego images are approximately 47 dB for gray level cover images and 43 dB for binary cover images. Therefore, proposed method ensures higher visual quality stego images for both gray level and binary cover images.

Improving secret capacity of the method without a loss in PSNR and analysis of lower prime modulus on the security of the method for limited threshold values are considered as future work.

Acknowledgements

This research was supported by the Research Fund of Karadeniz Technical University (Project No. 2010.112.009.1). The authors would like to thank the reviewers for their valuable comments.

References

Mustafa Ulutas got his B.Sc. in EE at Karadeniz Technical University in 1985. He earned both his M.Sc. and PhD degrees in EE at Texas Tech University in 1991 and 1994 respectively. He joined the Department of Computer Engineering at Karadeniz Technical University as an assistant professor from 1994 to 2002. He then joined Ondokuz Mayis University where served as the chairman of the Computer Engineering Department until he moved back to the Department of Computer Engineering at Karadeniz Technical University in 2009. He is interested in computer hardware as well as information security.

Guzin Ulutas got her B.Sc., M.Sc., PhD in Computer Engineering at Karadeniz Technical University in 2002, 2004 and 2012 respectively. She worked as a research assistant in the Department of Computer Engineering at Ondokuz Mayis University from 2005 to 2009. She became Assistant Professor in Computer Engineering Department at Karadeniz Technical University in 2012. Her main research interests are steganography and secret image sharing.

Vasif V. Nabiyev received B.Sc. and M.Sc. degrees in the Faculty of Computing Technologies and Informatics from St. Petersburg Electrotechnical University in 1985, and a PhD degree in the Department of Computer Science from Moscow Technical University in 1990. Then, in 2005, he received Professor degree in Computer Science Department from Karadeniz Technical University, Turkey, where he still lectures. He is currently the chairman of Department of Computer Engineering in Karadeniz Technical University. His research interests are in artificial intelligence, biometry, security, human computer interaction, image processing, natural language processing, operational research, discrete and applied mathematics, combinatorial algorithms, game theory, logic programming. He has published over 70 research papers, and authored three textbooks titled “Artificial Intelligence: Problems, Methods and Algorithms” (1st ed. 2003, 2nd ed. 2005, 3th ed. 2010), “Introduction of Algorithms” (2007) and, “Combinatorial Algorithms” (2007).