A Secret Sharing Based Method using Alpha Channel Hiding with A Data Repair Capability

Ms. Jufaira k
Electronics and Communication
KMCT College of Engineering
Kallanthode,Calicut, India

Sabitha V
Electronics and Communication
KMCT College of Engineering
Kallanthode, Calicut, India

***Abstract*— A secret sharing based method using alpha channel hiding with a data repair capability is proposed. Here a secret sharing method namely Shamir secret sharing scheme for gray scale document image via the use of portable network graphics (PNG) is used. An authentication signal is generated for each block of a grayscale document image, which, together with the binarized block content, is transformed into several shares using the Shamir secret sharing scheme. The involved parameters are carefully chosen so that as many shares as possible are generated and embedded into an alpha channel plane. An alpha channel is a plane that represents transparency information on a per pixel basis can be included in gray scale and true color PNG images. The alpha channel plane is then combined with the original grayscale image to form a PNG image. During the embedding process, the computed share values are mapped into a range of alpha channel values near their maximum value of 255 to yield a transparent stego-image with a disguise effect. In the process of image authentication, an image block is marked as tampered if the authentication signal computed from the current block content does not match that extracted from the shares embedded in the alpha channel plane. Data repairing is then applied to each tampered block by a reverse Shamir scheme after collecting two shares from unmarked blocks. Secret data is hided within the cover image and alpha channel plane; the data may be same or different. The secret message is encrypted before transmission for higher security purpose and the hidden data is retrieved at the receiving end. Good experimental results prove the effectiveness of the proposed method for real applications.**

Keywords*— *Data hiding, data repair, secret sharing, image authentication, Portable Network Graphics (PNG), double data hiding

I. INTRODUCTION

With the fast advance of digital technologies, it is easy to make visually imperceptible modifications to the contents of digital images. How to ensure the integrity and the authenticity of a digital image is thus a challenge. It is desirable to design effective methods to solve this kind of image authentication problem, particularly for images of documents whose security must be protected. It is also hoped that, if part of a document image is verified to have been illicitly altered, the destroyed content can be repaired. Such image content authentication and self-repair capabilities are useful for the security protection of digital documents in many fields, such as important certificates, signed documents etc., as main contents, are often digitized into grayscale images with two major gray values, one being of the background (including mainly blank spaces) and the other of the foreground (including mainly texts). It is noted that such images, although gray valued in nature, look like binary. It seems that such binary-like grayscale document images may be thresholded into binary ones for later processing, but such a thresholding operation often destroys the smoothness of the boundaries of text characters, resulting in visually unpleasant stroke appearances with zigzag contours. Therefore, in practical applications, text documents are often digitized and kept as grayscale images for later visual inspection.

Digital preservation is a formal endeavor to ensure that digital information of continuing value remains accessible and usable. It involves planning, resource allocation, and application of preservation methods and technologies In general, the image authentication problem is difficult for a binary document image because of its simple binary nature that leads to perceptible changes after authentication signals are embedded in the image pixels. Such changes will arouse possible suspicions from attackers. A good solution to such binary image authentication should thus take into account not only the security issue of preventing image tampering but also the necessity of keeping the visual quality of the resulting image. In this paper, propose an authentication method that deals with binary-like grayscale document images instead of pure binary ones and simultaneously solves the problems of image tampering detection and visual quality keeping. Several methods for binary image authentication have been proposed in the past. Wu and Liu manipulated the so-called flippable pixels to create specific relationships to embed data for authentication and annotation of binary images. Yang and Kot proposed a two-layer binary image authentication method in which one layer is used for checking the image fidelity and the other for checking image integrity. In the method, a connectivity- preserving transition criterion for determining the flippability of a pixel is used for embedding the cryptographic signature and the block identifier. Later, Yang and Kot proposed a

pattern-based data hiding method for binary image authentication in which three transition criteria are used to determine the flippabilities of pixels in each block, and the watermark is adaptively embedded into embeddable blocks to deal with the uneven embeddability condition in the host image. In the method proposed in H Y Kim, a set of pseudorandom pixels in a binary or halftone image are chosen and cleared, and authentication codes are accordingly computed and inserted into selected random pixels. In Tzeng and Tsai's method, randomly generated authentication codes are embedded into image blocks for use in image authentication, and a so-called code holder is used to reduce image distortion resulting from data embedding. Lee et al. proposed a Hamming-code-based data embedding method that flips one pixel in each binary image block for embedding a watermark, yielding small distortions and low false negative rates. Lee et al. improved the method later by using an edge line similarity measure to select flippable pixels for the purpose of reducing the distortion Double data hiding is used to hide secret messages within image data as well as the alpha channel created for the image. Thus we can get high security and double capacity for secret message transferring. Messages are encrypted before embedding in to the image by using one of the encryption key and they are decrypted after transmission at the reliever end. Here also get high security for the hidden messages.

II. SHAMIR METHOD FOR SECRET SHARING AND ALPHA CHANNEL

In this paper, a method for the authentication of document images with an additional self-repair capability for fixing tampered image data is proposed. The input cover image is assumed to be a binary-like grayscale image with two major gray values like the one shown in. After the proposed method is applied, the cover image is transformed into a stego-image in the Portable Network Graphics (PNG) format with an additional alpha channel for transmission on networks or archiving in databases. The stego-image, when received or retrieved, may be verified by the proposed method for its authenticity. Integrity modifications of the stego-image can be detected by the method at the block level and repaired at the pixel level. In case the alpha channel is totally removed from the stego-image, the entire resulting image is regarded as inauthentic, meaning that the fidelity check of the image fails. The proposed method is based on the so-called (k,n) -threshold secret sharing scheme proposed by Shamir in which a secret message is transformed into shares for keeping by participants, and when of the shares, not necessarily all of them, are collected, the secret message can be losslessly recovered. Such a secret sharing scheme is useful for reducing the risk of incidental partial data loss.

Conventionally, the concepts of "secret sharing" and "data hiding for image authentication" are two irrelevant issues in the domain of information security. However, in the proposed method, we combine them together to develop a new image authentication technique. The secret sharing scheme is used in the developed technique not only to carry authentication signals and image content data but also to help repair tampered data through the use of shares.

An issue in the self-repairing of tampered data at attacked image parts is that, after the original data of the cover image are embedded into the image itself for use in later data repairing, the cover image is destroyed in the first place and the original data are no longer available for data repairing, resulting in a contradiction. A solution to this problem is to embed the original image data somewhere else without altering the cover image itself. The way proposed in this paper to implement this solution is to utilize the extra alpha channel in a PNG image to embed the original image data. However, the alpha channel of the PNG image is originally used for creating a desired degree of transparency for the image. Moreover, embedding of data into the alpha channel will create random transparency in the resulting PNG image, producing an undesirable opaque effect. One way out, as proposed in this paper, is to map the resulting alpha channel values into a small range near their extreme value of 255, yielding a nearly imperceptible transparency effect on the alpha channel plane.

Another problem encountered in the self-repairing of the original image data is that the data to be embedded in the carrier are often large sized. For our case here with the alpha channel as the carrier, this is not a problem because the cover image that we deal with is essentially binary-like, and thus, we may just embed into the carrier a binary version of the cover image, which includes much less data. Furthermore, through a careful design of authentication signals, a proper choice of the basic authentication unit (i.e., the unit of 2×3 image block) and a good adjustment of the parameters in the Shamir scheme, we can reduce the data volume of the generated shares effectively so that more shares can be embedded into the alpha channel plane. It is noted that, by the proposed method, the larger the number of shares is, the higher the resulting data repair capability becomes, as shown in the subsequent sections. Finally, we distribute the multiple shares randomly into the alpha channel to allow the share data to have large chances to survive attacks and to thus promote the data repair capability. To the best of our knowledge, this is the first secret-sharing-based authentication method for binary-like grayscale document images. It is also the first authentication method for such document images through the use of the PNG image. Note that this method is not a secret-sharing technique but a document image authentication method.

In the (k,n)-threshold secret sharing method proposed by Shamir, secret d in the form of an integer is transformed into shares, which then are distributed to participants for them to keep; and as long as of the shares are collected, the original secret can be accordingly recovered, where $k \leq n$. The detail of the method is reviewed in the following.

A. *Algorithm 1: -threshold secret sharing.*

Input: secret d in the form of an integer, number n of participants, and threshold k≤n .
Output: n shares in the form of integers for the n participants to keep.
Step 1. Choose randomly a prime number p that is larger than d.
Step 2. Selectk-1 integer values $c_1, c_2 \ldots . c_{k-1}$within the range of 0 through p-1.
Step 3. Select n distinct real values $x_1, x_2 \ldots . x_n$.
Step 4. Use the following (k-1)-degree polynomial to compute n function values$F(x_i)$, called partial shares for i=1,2,…n, i.e.,

$$F(x_i) = (d + c_1 x_i + c_2 {x_i}^2 \ldots \ldots \ldots + c_{k-1} {x_i}^2)_{mod\ p} \quad (1)$$

Step 5. Deliver the two-tuple$(x_i, F(x_i))$ as a *share* to the ith participant where i=1,2,….n.

Since there are k coefficients, namely,d,and c_1 through c_{k-1} in (1) above, it is necessary to collect at least k shares from the n participants to form k equations of the form of (1) to solve these k coefficients in order to recover secret d . This explains the term threshold k for and the name(k,n) - threshold for the Shamir method. Below is a description of the just-mentioned equation-solving process for secret recovery.

B. Algorithm 2: Secret recovery.

Input: k shares collected from the participants and the n prime number p with both k and p being those used in Algorithm 1.
Output: secret d hidden in the shares and coefficients c_i used in (1) in algorithm 1, where i=1,2,….,k
Steps.
Step1. Use the k shares
$(x_1, F(x_1)), (x_2, F(x_2)), \ldots \ldots , (x_k, F(x_k))$ to

$$F(x_j) = (d + c_1 x_j + c_2 {x_j}^2 + \ldots \ldots + c_{k-1} {x_j}^{k-1})_{mod\ p}$$

$$Where\ j = 1,2 \ldots . k.$$

Step2. Solve y=the k equations above by Lagrange's interpolation to obtain d as follows:

$$d = (-1)^{k-1}[F(x_1)\frac{x_2 x_3 \ldots\ldots\ldots x_k}{(x_1-x_2)(x_1-x_3)\ldots\ldots(x_1-x_k)} + F(x_2)\frac{x_1 x_3 \ldots\ldots\ldots x_k}{(x_2-x_1)(x_2-x_3)\ldots\ldots(x_2-x_k)} + \ldots\ldots + F(x_k)\frac{x_1 x_2 \ldots\ldots\ldots x_{k-1}}{(x_k-x_1)(x_k-x_2)\ldots\ldots(x_k-x_{k-1})}]_{mod\ p}$$

Step3. Compute c_1 through c_{k-1} by expanding the following equality and comparing the result with (2) in step 1 while regarding variable *x* in the equality below to be x_j in (2):

$$F(x) = [F(x_1)\frac{(x-x_2)(x-x_3)\ldots\ldots\ldots(x-x_k)}{(x_1-x_2)(x_1-x_3)\ldots\ldots(x_1-x_k)} + F(x_2)\frac{(x-x_1)(x-x_3)\ldots\ldots\ldots(x-x_k)}{(x_2-x_1)(x_2-x_3)\ldots\ldots(x_2-x_k)} + \ldots\ldots\ldots + F(x_k)\frac{(x-x_1)(x-x_2)\ldots\ldots\ldots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\ldots\ldots(x_k-x_{k-1})}]_{mod\ p}$$

Step 3 in the above algorithm is additionally included for the purpose of computing the values of parameters c_i in the proposed method. In other applications, if only the secret value d need be recovered, this step may be eliminated.

C. *Alpha channel*

The concept of an alpha channel was introduced by Alvy Ray Smith in the late 1970s, and fully developed in a 1984 paper by Thomas Porter and Tom Duff. The alpha channel is a color component that represents the degree of transparency (or opacity) of a color (i.e., the red, green and blue channels). It is used to determine how a pixel is rendered when blended with another. The alpha channel controls the transparency or opacity of a color. Its value can be represented as a real value, a percentage, or an integer: full transparency is 0.0, 0% or 0, whereas full opacity is 1.0, 100% or 255, respectively. When a color (source) is blended with another color (background), e.g., when an image is overlaid onto another image, the alpha value of the source color is used to determine the resulting color. If the alpha value is opaque, the source color overwrites the destination color; if transparent, the source color is invisible, allowing the background color to show through. If the value is in between, the resulting color has a varying degree of transparency/opacity, which creates a translucenteffect.

The alpha channel is used primarily in alpha blending and alpha compositing. alpha compositing is the process of combining an image with a background to create the appearance of partial or full transparency. It is often useful to render image elements in separate passes, and then combine the resulting multiple 2D images into a single, final image called the composite. For example, compositing is used extensively when combining computer-rendered image elements with live footage.

D. *Creation of alpha channel for cover image*

Here one alpha channel is created for one cover image. First of all, pixels from the cover images are separated. These pixels are converted to shares by Shamir secret sharing scheme .these shares are embedded in the alpha channel plane, which is directly created from the software, as row column basis.

Alpha channel is transmitted along with the image so that which is reconstructed at the receiver end we can reconstruct the original image if there is any tampering occurs.

III. IMAGE AUTHENTICATION , DATA REPAIRING AND DOUBLE DATA HIDING

In the proposed method, a PNG image is created from a binary- type grayscale document image I with an alpha channel plane. The original image I may be thought as a grayscale channel plane of the PNG image. An illustration of this process of PNG image creation is shown in Fig. 1. Next, I is binarized by moment-preserving thresholding, yielding a binary version of I , which we denote as I_b . Data for authentication and repairing are then computed from I_b and taken as input to the Shamir secret

sharing scheme to generate secret shares. The share values are subsequently mapped into a small range of alpha channel values near the maximum transparency value to create an imperceptibility effect. Finally, the mapped secret shares are randomly embedded in to the alpha channel for the purpose of promoting the security protection and data repair capabilities. Two block diagrams describing the proposed method are shown in Figs. 2 and 3. Since the alpha channel plane is used for carrying data for authentication and repairing, no destruction will occur to the input image in the process of authentication. In contrast, conventional image authentication methods often sacrifice part of image contents, such as least significant bits (LSBs) or flappable pixels, to accommodate data used for authentication. In addition, once a stego-image generated from a conventional method such as an LSB-based one is unintentionally compressed by a lossy compression method, the stego-image might cause false positive alarms in the authentication system. In contrast, the proposed method yields a stego image in the PNG format, which, in normal cases, will not be further compressed, reducing the possibility of erroneous authentication caused by imposing undesired compression operations on the stego-image. A detailed algorithm is proposed for describing the generation of a stegoimage in the PNG format.

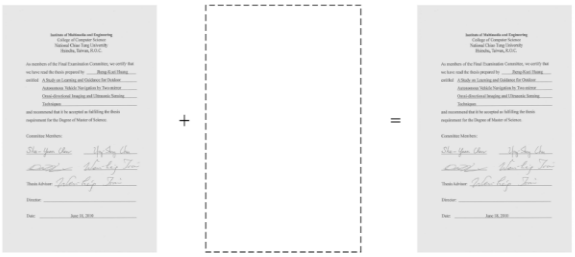

Fig.1. Illustration of creation of a gray scale document image and an additional channel plane

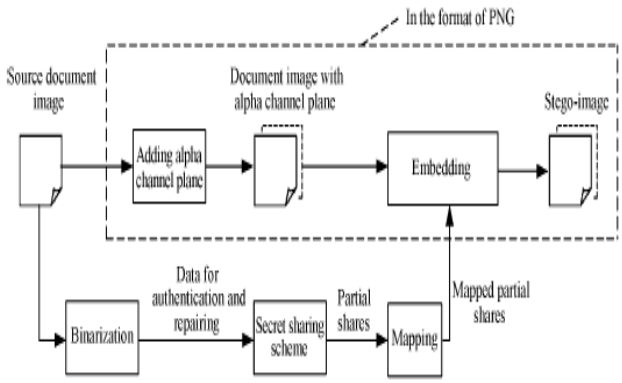

Fig.2. Illustration of creating a PNG image from a gray scale document image and an alpha channel plane

A. *Double data hiding*

The transmitting data consists of an image or document with an added alpha channel. Double data hiding gives data hiding on both image and alpha channel. Image consists of pixels, such that data is hided on these pixels. Number of pixels used for hiding depends on the length of the message. First of all for the first bit of the message, beginning pixel of the image is converted to binary. The image is of a matrix form, for this hiding which is changed to array format. The last digits of the binary values are used for the message hiding. So that the first bit of the message is hided on the last bits of binary values of the first pixel of the image which is to be transmitted. The process is repeated until the message is completed. After the completion of the message, the process is automatically broken.

The exact reverse process is take place at the end receiver. Message retrieving is the receiver end application. Arrays are converted to matrix and the secret message is retrieved from the last bits of each pixels.

B. *Data hiding by encryption*

Data hiding along with encryption and decryption of the data to be transmitted. The secret data is encrypted before which is embedded on to the image and at the receiving side, the data is decrypted. One of the method with same process are used for both encryption and decryption.

C. *Merits of the Proposed Method*

In addition to being capable of data repairing and being blind in nature (requiring no overhead other than the stego-image), the proposed method has several other merits, which are described in the following.

- Double capacity is achieved as the secret message is hided within the image and in the alpha channel plane.
- High security by providing encryption to the message before embedding to the image. Decryption is required to retrieve the message.
- Data hiding along with the alpha channel plane is one of the advantages of this method.

  Providing pixel-level repairs of tampered image parts—As long as two untampered partial shares can be collected, a tampered block can be repaired at the pixel level by the proposed method. This yields a better repair effect for texts in images because text characters or letters are smaller in size with many curved strokes and need finer pixel-level repairs when tampered with.
- Having higher possibility to survive image content attacks— By skillfully combining the Shamir scheme, the authentication signal generation, and the random embedding of multiple shares, the proposed method can survive malicious attacks of common content modifications, such as superimposition, painting,

etc., as will be demonstrated by experimental results subsequently described.

- Making use of a new type of image channel for data hiding—Different from common types of images, a PNG image has the extra alpha channel plane that is normally used to produce transparency to the image. It is differently utilized by the proposed method for the first time as a carrier with a large space for hiding share data. As a comparison, many other methods use LSBs as the carriers of hidden data.

- Causing no distortion to the input image—Conventional image authentication methods that usually embed authentication signals into the cover image itself will unavoidably cause destruction to the image content to a certain extent. Different from such methods, the proposed method utilizes the pixels' values of the alpha channel for the purpose of image authentication and data repairing, leaving the original image (i.e., the grayscale channel) untouched and thus causing no distortion to it. The alpha channel plane may be removed after the authentication process to get the original image. Fig. 5 shows the framework of the proposed method in this aspect, and Fig. 6, shown for comparison, illustrates a conventional image authentication method.

- Enhancing data security by secret sharing—Instead of hiding data directly into document image pixels, the proposed method embeds data in the form of shares into the alpha channel of the PNG image. The effect of this may be regarded as double-fold security protection, one fold contributed by the shares as a form of disguise of the original image data and the authentication signals and the other which is created to be nearly transparent, as previously mentioned.

D. *Measures for security enhancement*

The secret key K , which is used to randomize the pixel positions for embedding the mapped partial shares through mentioned in Step 9 of Algorithm 3, provides a measure to protect the shares. More specifically, as described in Algorithm 3, each block in the alpha channel plane may be regarded to consist of two parts, i.e., the first part including the first two pixels and the second including the remaining four. The first part of each block is used for keeping the first two partial shares and , and the second part for keeping the last four partial shares through of other blocks located at random positions. Therefore, the probability of correctly guessing the locations of all the embedded partial shares in a stego-image is$1/[(m \times n) - (m \times {}^{n}/_{6}) \times 2]$, where m×n is the size of the cover image, m×$n/6$ is the total number of blocks, each with six pixels, $(m \times n) - (m \times {}^{n}/_{6}) \times 2$ and is the total number of pixels in the blocks other than those in the first parts of all the blocks. This probability is obviously very small for common image sizes, meaning that a correct guess of the embedded partial shares is nearly impossible.

To enhance further the security of the data embedded in the stego-image, one additional measure is adopted in the proposed method (but not included in the previously proposed algorithms for clarity of algorithm descriptions). It is the randomization of the constant values of x_1 through x_6 used in Step 6 of Algorithm3 and Step 3(2) in Algorithm 4. Specifically, in Step 3(2) in Algorithm 4, we can see that the input shares into Algorithm 2, i.e.$(1, q_1)$, and $(2,q_2)$, can be easily forged, leading to the possibility of creating fake authentication signals. To remedy this weakness, with the help of another secret key, we may choose these values of x_1 through x_6 for each block to be random within the allowed integer range of $0 \leq x_i \leq$ p (=17). Then, the probability of correctly guessing all these values for all the m×$n/6$ blocks in a stego-image can be figureout to be $1/[(17 \times 16 \times 15 \times 14 \times 13 \times 12)]^{m \times n/6} \approx 1/(8.911 \times 10^6)^{m \times n/6}$, which is also very small for common image sizes m× n.

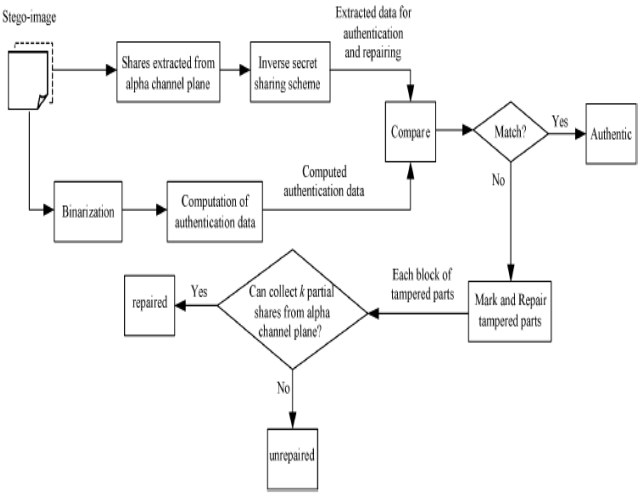

Fig.3:. Authentication process including verification and self-repairing of a stego-image in PNG format.

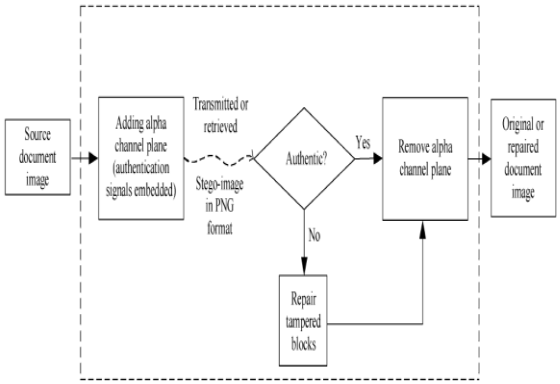

Fig.4: Framework of proposed document image authentication method

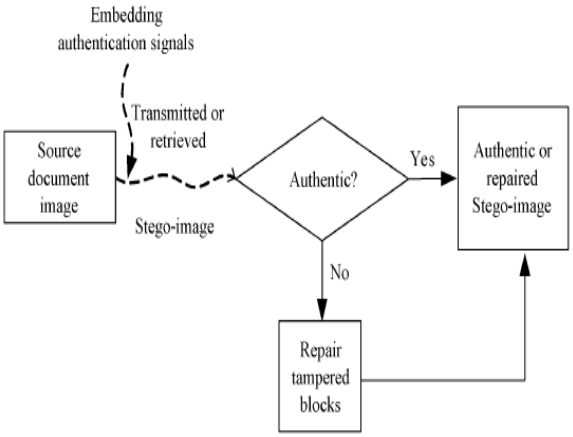

Fig.5: Framework of conventional image authentication method

IV. PROPOSED ALGORITHM

A. *Algorithm for Generation of a Stego-Image*

A detailed algorithm for describing the generation of a stego image in the PNG format of the proposed method is presented in the following.

Algorithm 3: Generation of a stego-image in the PNG format from a given gray scale image.
Input: A gray scale document image I with two major gray values and a secret key k
Output: Stego image I' in the PNG with relevant data embedded, including the authentication signals and the data used for repairing.
Steps.
Stage I—generation of authentication signals.
Step 1. (Input image binarization) Apply moment-preserving thresholding [13] to to obtain two representative gray values g_1 and g_2 , compute $T = {g_1 + g_2}/{2}$, and use as a threshold to binarize I, yielding a binary version $I_b$with "0" representing g_1 and "1" representing g_2 .
Step 2. (Transforming the cover image into the PNG format)
Transform I into a PNG image with an alpha channel plane I_α by creating a new image layer with 100% opacity and no color as I_α and combining it with I using an image processing software package.
Step 3. (Beginning of looping) Take in an unprocessed raster-scan order a 2× 3 block of with pixels $p_1, p_2 \ldots \ldots p_6$
.Step 4. (Creation of authentication signals) Generate a 2-bit authentication signal $s = a_1a_2$ with $a_1 = P_1XORp_2XORp_3$and $a_1 = p_4XORp_5XORp_6$, where denotes the exclusive-or operation.
Stage II—creation and embedding of shares.

Step 5. (Creation of data for secret sharing) Concatenate the 8 bits of a_1, a_2 , and p_1 through $p_6$to form an 8-bit string, divide the string into two 4-bit segments, and transform the segments into two decimal numbers m_1 and m_2 , respectively.

Step 6. (Partial share generation) Set p ,c_i , and x_i in (1) of Algorithm 1 to be the following values: 1) p=17 (the smallest prime number larger than 15); 2)$d = m_1$ and $c_1 = m_2$; and 3)$x_1 = 1, x_2 = 2 \ldots . x_6 = 6$. Perform Algorithm 1 as a (2, 6)-threshold secret sharing scheme to generate six partial shares through using the following equations:

$$q_i = F(x_i) = (d + c_1 x_i)_{\mathrm{mod}\ p} \quad (3)$$

where i=1,2,…6
Step 7. (Mapping of the partial shares) Add 238 to each of $q_1$through q_6, resulting in the new values of q_1' through q_6' , respectively, which fall in the nearly total transparency range of 238 through 254 in the alpha channel planeI_α .
Step 8. (Embedding two partial shares in the current block) Take block B_α in I_α corresponding to B_b in I_b , select the first two pixels in B_α in the raster-scan order, and replace their values by q_1' and q_2' , respectively.
Step 9. (Embedding remaining partial shares at random pixels)
Use key to select randomly four pixels in I_α but outside B_α, which are unselected yet in this step, and not the first two pixels of any block; in the raster-scan order, replace the four pixels' values by the remaining four partial shares q_3' through q_6' generated above, respectively.
Step 10. (End of looping) If there exists any unprocessed block in I_b, then go to Step 3; otherwise, take the final I in the PNG format as the desired stego-image I'.

The possible values of $q_1$through q_6 yielded by (3) above are between 0 and 16 because the prime number pused there is 17. After performing Step 7 of the above algorithm, they become q_1' through q_6' , respectively, which all fall into a small interval of integers ranging from 238 to 254 with a width of 17 (the value of the prime number). The subsequent embedding of q_1' through q_6' in such a narrow interval into the alpha channel plane means that very similar values will appear everywhere in the plane, resulting in a nearly uniform transparency effect, which will not arouse notice from an attacker.

The reason why we choose the prime number to be 17 in the above algorithm is explained here. If it was instead chosen to be larger than 17, then the aforementioned interval will be enlarged, and the values of q_1' through q_6' will become possibly smaller than 238, creating an undesired less transparent but visually whiter stego-image. On the other hand, the 8 bits mentioned in Steps 5 and 6 above are transformed into two decimal numbers m_1 and m_2 with their maximum values being 15 (see Step 5 above), which are constrained to lie in the range of 0 through p-1 (see Step 2 in Algorithm 1). Therefore, p should not be chosen to be smaller than 16. In short, p=17 is an optimal choice.

As to the choice of the block size, the use of a larger block size, such as 2 4 or 3 3, will reduce the precision of the resulting integrity authentication (i.e., the stego-image will be verified in a spatially coarser manner). On the other hand, it seems that a smaller block size such as 2 × 2 instead of 2 × 3 may be tried to increase the

authentication precision. However, a block in the alpha channel with a size of 2 × 2 can be used to embed only four partial shares instead of six (see Steps 6–9 of Algorithm 3). This decreases the share multiplicity and thus reduces the data repair capability of the method. In short, there is a tradeoff between the authentication precision and the data repair capability, and our choice of the block size of 2 × 3 is a balance in this aspect. Finally, we use Fig. 4 to illustrate Steps 8 and 9 of Algorithm 3, where a core idea of the proposed method is presented, i.e., two shares of the generated six are embedded at the current block and the other four are embedded at four randomly selected pixels outside the block, with each selected pixel not being the first two ones in any block.

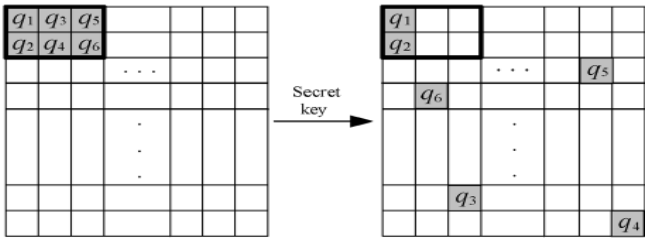

Fig.6: Illustration of embedding six shares created for a block: Two shares embedded at the current block, and the other four in four randomly selected pixels outside the block, with each selected pixel not being the first two ones in any block.

B. *Algorithm for Stego-Image Authentication*

A detailed algorithm describing the proposed stego-image authentication process, including both the verification and the self-repairing of the original image content, is presented in the following.

Algorithm 4: Authentication of a given stego-image in the PNG format.
Input: stego-image I' , the representative gray values g_1 and g_2 , and the secret key k used in Algorithm 3.
Output: image I_r with tampered blocks marked and their data
repaired if possible.
Stage I—extraction of the embedded two representative gray values.
Step 1. (Binarization of the stego-image) Compute $T = (g_1 + g_2)/2$, and use it as a threshold to binarize I' , yielding a binary version I'_b of I' with "0" representing g_1 and "1" representing g_2 .
Stage II—verification of the stego-image.
Step 2. (Beginning of looping) Take in a raster-scan order an unprocessed block B'_b from I'_b with pixel values p_1 through p_6 , and find the six pixels' values q'_1 through q'_6 of the corresponding block B'_α in the alpha channel plane I'_α of I' .
Step 3. (Extraction of the hidden authentication signal)
Perform the following steps to extract the hidden 2-bit authentication signal $s = a_1a_2$ from B'_α :
(1) Subtract 238 from each of q'_1 and q'_2 to obtain two partial shares q_1 and q_2 of B'_b, respectively.
(2) With shares $(1, q_1)$ and $(1, q_2)$ as input, perform Algorithm 2 to extract the two values d and c_1(the secret and the first coefficient value, respectively) as output.
1) (3) Transform d and c_1 into two 4-bit binary values, concatenate them to form an 8-bit string S, and take the first 2 bits a_1 and a_2 of S to compose the hidden authentication signal $s = a_1a_2$.
Step 4. (Computation of the authentication signal from the current block content) Compute a 2-bit authentication signal $s' = a'_1a'_2$ from values p_1 through p of the six pixels of B'_b by $a'_1 = p_1XORp_2XORp_3$ and $a'_2 = p_4XORp_5XORp_6$.
Step 5. (Matching of the hidden and computed authentication signals and marking of tampered blocks) Match s and s' by checking if $a_1 = a'_1$ and $a_2 = a'_2$, and if any mismatch occurs,

mark B'_b , the corresponding block B' in I' , and all the partial shares embedded in B'_α as tampered.
Step 6. (End of looping) If there exists any unprocessed block in I'_b , then go to Step 2; otherwise, continue.

Stage III—self-repairing of the original image content
Step 7. (Extraction of the remaining partial shares) For each block B'_α in I'_α , perform the following steps to extract the remaining four partial shares q_3 through q_6 of the corresponding block B'_b in I'_b from blocks in other than B'_α.
(1) Use key k to collect the four pixels in I'_α in B'_b the same order as they were randomly selected for in Step 9 of Algorithm 3, and take out the respective data q'_3, q'_4, q'_5 and q'_6 embedded in them.
(2) Subtract 238 from each of q'_3 through q'_6 to obtain q_3 through q_6, respectively.
Step 8. (Repairing the tampered regions) For each block B' in I' marked as tampered previously, perform the following steps to repair it if possible.
(1) From the six partial shares q_1 through q_6 of block B'_b in I'_b corresponding to B' (two computed in Step 3(1) and four in Step 7(2) above), choose two of them, e.g. q_k, and q_l, which are not marked as tampered, if possible.
(2) With shares (k, q_k) and (l, q_l) as input, perform Algorithm 2 to extract the values of d and c_1 (the secret and the first coefficient value, respectively) as output.
(3) Transform d and c_1 into two 4-bit binary values, and concatenate them to form an 8-bit string S'
(4) Take the last 6 bits $b'_1, b'_2, \ldots b'_6$ from S', and check their binary values to repair the corresponding tampered pixel values $y'_1, y'_2, \ldots y'_6$ of block B' by the following way:
If $b'_i = 0$, set $y'_i = g_1$; otherwise, set $y'_i = g_2$
where i=1,2,..6.
Step 9. Take the final I' as the desired self-repaired image I_r

C. *Double data hiding*

Deals with hiding on both cover image and alpha channel plane

1) Hiding on cover image

Step 1(Integer to binary conversion) Pixel integer value of each pixel in the cover image is converted to binary values.

Step 2(Message embedding) Placing each message to the end binary values so that the least significant bits are covered with secret messages. The number of bits considering for this is based upon the message length.

Step 3(Conversion to png format) Converting the secret message embedded pixel in to png format for transmission purpose. At the receiver the message is retrieved.

2) Hiding on alpha channel

The exact same process as the above take place but the secret messages is hided within the shares of the alpha channel plane. The message retrieving at the end receiver is the reverse process of transmitting side.

CONCLUSION

A new blind image authentication method with a data repair Capability for binary-like grayscale document images based on secret sharing has been proposed with double data hiding and by encryption. Both the generated authentication signal and the content of a block have been transformed into partial shares by the Shamir method, which have been then distributed in a well-designed manner into an alpha channel plane to create a stego-image in the PNG format.

Secret messages are hided within cover images and alpha channel plane thus will get double capacity for transmission. For the self-repairing of the content of a tampered block, the reverse Shamir scheme has been used to compute the original content of the block from any two untampered shares. Measures for enhancing the security of the data embedded in the alpha channel plane have been also proposed.

REFERENCES

[1] Che-Wei LEE and Wen-Hsiang Tsai, " A secret sharing based image authentication of gray scale document image via the use of PNG image with a data repair capabolity", IEEE, VOL.21, Jan 2012.

[2] H. Yang and A. C. Kot, "Pattern-based data hiding for binary images authentication by connectivity-preserving," IEEE Trans. Multimedia, vol. 9, no. 3, pp. 475–486, Apr. 2007

[3] C. H. Tzeng and W. H. Tsai, "A new approach to authentication of binary images for multimedia communication with distortion reduction and security enhancement," IEEE Commun. Lett., vol. 7, no. 9, pp. 443–445, Sep. 2003.

[4] M. Wu and B. Liu, "Data hiding in binary images for authentication and annotation," IEEE Trans. Multimedia, vol. 6, no. 4, pp. 528–538, Aug. 2004.

[5] H. Y. Kim "Secure authentication watermarking for halftone and binary images," Int. J. Imag. Syst. Technol., vol. 14, no. 4, pp. 147–152, 2004.

[6] C. C. Lin and W. H. Tsai, "Secret image sharing with steganography and authentication," *J. Syst. Softw.*, vol. 73, no. 3, pp. 405–414, Nov./ Dec.2004