A Secret-Sharing-Based Method for Authentication of Grayscale Document Images via the Use of the PNG Image with a Data Repair Capability

Che-Wei Lee, *Student Member*, *IEEE* and Wen-Hsiang Tsai, *Senior Member*, *IEEE*

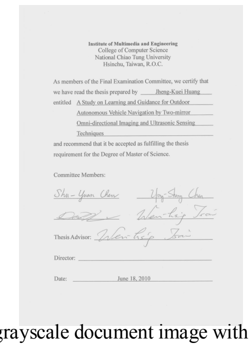
***Abstract*—A new blind authentication method based on the secret sharing technique with a data repair capability for grayscale document images via the use of the PNG image is proposed. An authentication signal is generated for each block of a grayscale document image, which, together with the binarized 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. 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. Measures for protecting the security of the data hidden in the alpha channel are also proposed. Good experimental results prove the effectiveness of the proposed method for real applications.**

***Index Terms*—Image authentication, grayscale document image, secret sharing, data hiding, PNG image, data repair.**

I. Introduction

Digital image is a form for preserving important information. However, 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 authenticity of a digital image is thus a challenge. It is desirable to design effective methods to solve this kind of *image authentication* problem [1]–[3], especially 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 altered illicitly, the destructed content can be repaired. Such *image content authentication and self-repair capabilities* are useful for security protection of digital documents in many fields, such as important certificates, signed documents, scanned checks, circuit diagrams, art drawings, design drafts, last will and testaments, and so on.

Document images, which include texts, tables, line arts, 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, though gray-valued in nature, look like binary. For example, the two major gray values in the document image shown in Fig. 1 are 174 and 236, respectively. It seems that such *binary-like* grayscale document images may be thresholded into binary ones for later processing, but such a thresholding operation often destructs 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.



Institute of Multimedia and Engineering
College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

As members of the Final Examination Committee, we certify that we have read the thesis prepared by Jheng-Kuei Huang entitled A Study on Learning and Guidance for Outdoor Autonomous Vehicle Navigation by Two-mirror Omni-directional Imaging and Ultrasonic Sensing Techniques and recommend that it be accepted as fulfilling the thesis requirement for the Degree of Master of Science.

Committee Members:

Shu-Yuan Chen [illegible]

[illegible] [illegible]

Thesis Advisor: [illegible]

Director:

Date: June 18, 2010

Fig. 1. A binary-like grayscale document image with two major gray values.

In general, the image authentication problem is difficult for a binary document image because of its simple binary nature which 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 thus should 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 study, we propose an authentication method which deals with *binary-like* grayscale document images instead of *pure binary* ones, and solves simultaneously 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 [4] manipulated the so-called flippable pixels to create specific relationships to embed data for authentication and annotation of binary images. Yang and Kot [5] proposed a two-layer binary image authentication

This work was supported financially by the NSC project No. 99-2631-H-009-001.

C. W. Lee, Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: paradiserlee@gmail.com).

W. H. Tsai, Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: whtsai@cis.nctu.edu.tw); also with the Department of Information Communication, Asia University, Taichung 41354, Taiwan.

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 block identifier. Later, Yang and Kot [6] 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 by Kim et al. [7], a set of pseudo-random pixels in a binary or halftone image are chosen and cleared, and authentication codes are computed accordingly and inserted into selected random pixels. In Tzeng and Tsai's method [8], 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. [9] proposed a Hamming-code-based data embedding method which flips one pixel in each binary image block for embedding a watermark, yielding small distortions and low false negative rates. Lee et al. [10] improved the method later by using an edge line similarity measure to select flippable pixels for the purpose of reducing the distortion.

In this study, a method for 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 Fig. 1. After the proposed method is applied, the cover image is transformed into a *stego-image* in the 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 that 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 [11] in which a secret message is transformed into n *shares* for keeping by n participants; and when k of the n shares, not necessarily all of them, are collected, the secret message can be recovered *losslessly*. 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. But 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 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 destructed 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 study 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 used originally for creating a desired degree of transparency for the image. And 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 study, 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 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 we deal with is essentially binary-like, and so 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 Shamir's 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 can be seen 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 so to 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.

The remainder of this paper is organized as follows. In Section 2, the Shamir method on which the proposed method is based is reviewed first. In Section 3, the details of the proposed method, including authentication signal generation, share data embedding, and tampered data repairing are described. In Section 4, some discussions about the merits of the proposed method and possible enhancements of security protection are given. Experimental results and a comparison of performances of the proposed method with others are shown in Section 5, followed by conclusions in Section 6.

II. Review of Shamir's Method for Secret Sharing

In the (k, n)-threshold secret sharing method proposed by Shamir [11], a secret d in the form of an integer is transformed into shares which then are distributed to n participants to keep; and as long as k of the n shares are collected, the original secret can be recovered accordingly, where $k \leq n$. The detail of the method is reviewed in the following.

Algorithm 1: (k, n)-threshold secret sharing.

Input: a secret d in the form of an integer, the number n of participants, and a threshold $k \le n$.

Output: n shares in the form of integers for the n participants to keep.

Step 1. Choose randomly a prime number p which is larger than d.

Step 2. Select $k - 1$ 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 $i = 1, 2, \ldots, n$:

$$F(x_i) = (d + c_1x_i + c_2x_i^2 + \ldots + c_{k-1}x_i^{k-1})_{\mathrm{mod}\,p}, \qquad (1)$$

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

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 Eqs. (1) to solve these k coefficients in order to recover the secret d. This explains the term, *threshold*, for k and the name, (k, n)-*threshold*, for the Shamir method [11]. Below is a description of the just-mentioned equation-solving process for secret recovery.

Algorithm 2:* secret *recovery.

Input: k shares collected from the n participants and the prime number p with both k and p being those used in Algorithm 1.

Output: the secret d hidden in the shares and the coefficients c_i used in Eqs. (1) in Algorithm 1, where $i = 1, 2, \ldots, k - 1$.

Steps.

Step 1. Use the k shares $(x_1, F(x_1)), (x_2, F(x_2)), \ldots, (x_k, F(x_k))$ to set up the following equations:

$$F(x_j) = (d + c_1x_j + c_2x_j^2 + \ldots + c_{k-1}x_j^{k-1})_{\mathrm{mod}\,p}, \qquad (2)$$

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

Step 2. Solve the k equations above by Lagrange's interpolation to obtain d as follows [12]:

$$d = (-1)^{k-1}[F(x_1)\frac{x_2x_3...x_k}{(x_1-x_2)(x_1-x_3)...(x_1-x_k)} + F(x_2)\frac{x_1x_3...x_k}{(x_2-x_1)(x_2-x_3)...(x_2-x_k)} + ... + F(x_k)\frac{x_1x_2...x_{k-1}}{(x_k-x_1)(x_k-x_2)...(x_k-x_{k-1})}]_{\mathrm{mod}\,p}.$$

Step 3. Compute c_1 through c_{k-1} by expanding the following equality and comparing the result with (2) in Step 1 while regarding the variable x in the equality below to be x_j in (2):

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

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

III. Image Authentication and Data Repairing

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. 2. Next, I is binarized by moment-preserving thresholding [13], yielding a binary version of I, which we denote as I_b. Data for authentication and repairing then are computed from I_b and taken as input to Shamir's secret sharing scheme to generate n secret shares. The share values are mapped subsequently 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 into 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. 3 and 4.

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 LSBs or flippable pixels, to accommodate data used for authentication. In addition, once a stego-image generated from a conventional method like 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 compressed further, reducing the possibility of erroneous authentication caused by imposing undesired compression operations on the stego-image.

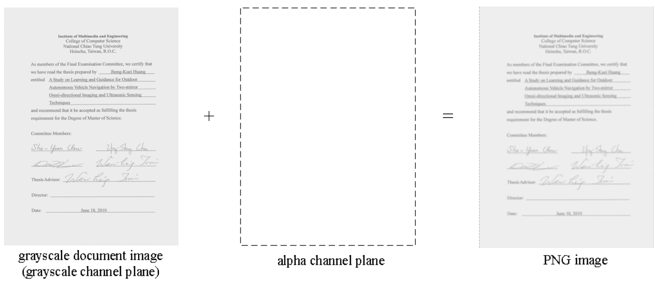


Fig. 2. Illustration of creation of a PNG image from a grayscale document image and an additional alpha channel plane.

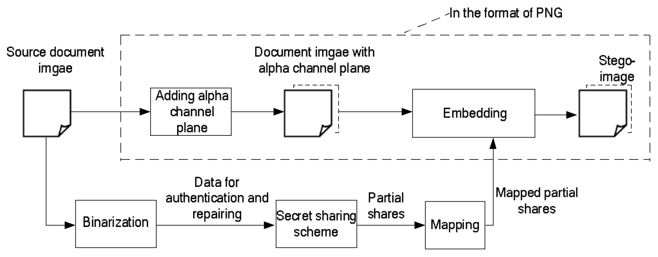


Fig. 3. Illustration of creating a PNG image from a grayscale document image and an alpha channel.

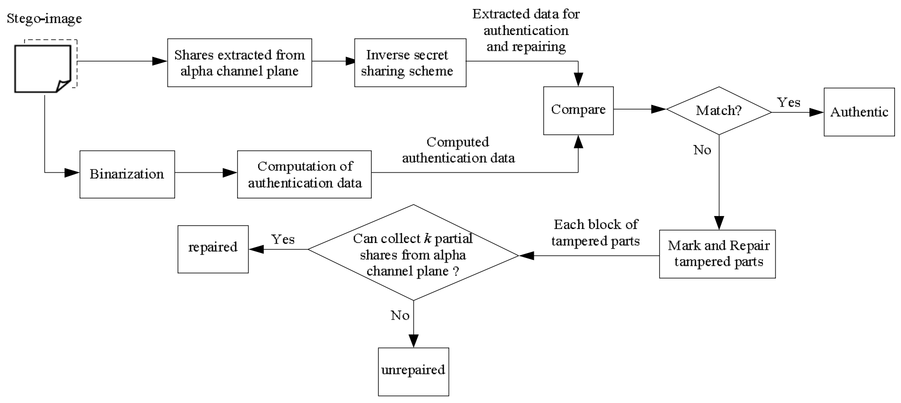

Stego-image
Shares extracted from alpha channel plane
Inverse secret sharing scheme
Extracted data for authentication and repairing
Compare
Match?
Yes
Authentic
No
Binarization
Computation of authentication data
Computed authentication data
Mark and Repair tampered parts
Each block of tampered parts
Can collect k partial shares from alpha channel plane ?
Yes
repaired
No
unrepaired

Fig. 4 Authentication process including verification and self-repairing of a stego-image in PNG format.

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 grayscale image.

Input: a grayscale document image I with two major gray values, and a secret key K.

Output: a stego-image I' in the PNG format 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 I to obtain two representative gray values g_1 and g_2; compute $T = (g_1 + g_2)/2$; and use T 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 B_b of I_b with pixels p_1, p_2, …, p_6.

Step 4. (*Creation of authentication signals*) Generate a 2-bit authentication signal $s = a_1a_2$ with $a_1 = p_1\oplus p_2\oplus p_3$ and $a_2 = p_4\oplus p_5\oplus p_6$ where $\oplus$ denotes the exclusive-OR operation.

Stage II --- creation and embedding of shares.

Step 5. (*Creation of data for secret sharing*) Concatenate the eight 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 Eqs. (1) of Algorithm 1 to be the following values: (a) p = 17 (the smallest prime number larger than 15); (b) $d = m_1$, $c_1 = m_2$; (c) x_1 = 1, x_2 = 2, …, x_6 = 6; and perform Algorithm 1 as a (2, 6)-threshold secret sharing scheme to generate six partial shares q_1 through q_6 using the following equations:

$$q_i = F(x_i) = (d + c_1x_i)_{\text{mod } p}, \tag{3}$$

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 plane I_α.

Step 8. (*Embedding two partial shares in the current block*) Take the 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 the key K 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; and 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 Eqs. (3) above are between 0 and 16 because the prime number p used 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). 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 chosen instead to be larger than 17, then the above-mentioned 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, like 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 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 through 9 of Algorithm 3). This decreases share multiplicity and so 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 2×3 is a *balance* in this aspect.

Finally, we use Fig. 5 to illustrate Steps 8 and 9 of Algorithm 3 where a core idea of the proposed method is presented: two shares of the generated six are embedded at the current block and the other four embedded at four randomly-selected pixels outside the block, with each selected pixel not being the first two ones in any block.

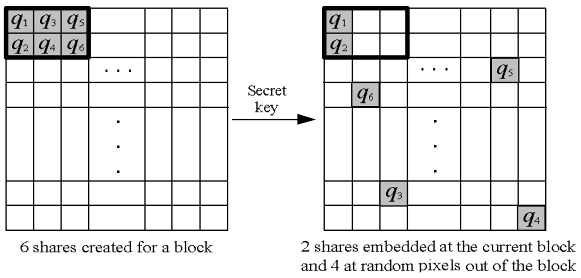

q_1 q_3 q_5
q_2 q_4 q_6
Secret key
q_1
q_2
q_5
q_6
q_3
q_4
6 shares created for a block
2 shares embedded at the current block and 4 at random pixels out of the block

Fig. 5. 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 verification and 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: a stego-image I'; the representative gray values g_1 and g_2, and the secret key K used in Algorithm 3.

Output: an 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 the shares $(1, q_1)$ and $(2, q_2)$ as input, perform Algorithm 2 to extract the two values d and c_1 (the secret and the first coefficient value, respectively) as output.

(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 two 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 two-bit authentication signal $s' = a_1'a_2'$ from the values p_1 through p_6 of the six pixels of B_b' by $a_1' = p_1 \oplus p_2 \oplus p_3$ and $a_2' = p_4 \oplus p_5 \oplus p_6$.

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 I_α' other than B_α'.

(1) Use the key K to collect the four pixels in I_α' in the same order as they were randomly selected for B_b' 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 the 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, say q_k and q_l, which are *not* marked as tampered, if possible.

(2) With the 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) 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 six bits b_1', b_2', …, b_6' from S' and check their binary values to repair the corresponding tampered pixel values y_1', y_2', …, 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.

IV. Discussions

A. 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.

(1) *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.

(2) *Having higher possibility to survive image content attacks* --- By combining skillfully the Shamir scheme, authentication signal generation, and 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 described subsequently.

(3) *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 which normally is used to produce transparency to the image. It is utilized differently 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.

(4) *Causing no distortion to the input image* --- Conventional image authentication methods which 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 so causing *no distortion* to it. The alpha channel plane may be removed after the authentication process to get the original image. Fig. 6 shows the framework of the proposed method in this aspect; and Fig. 7, shown for comparison, illustrates a conventional image authentication method.

(5) *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 fold contributed by the use of the alpha channel plane which is created to be nearly transparent, as mentioned previously.

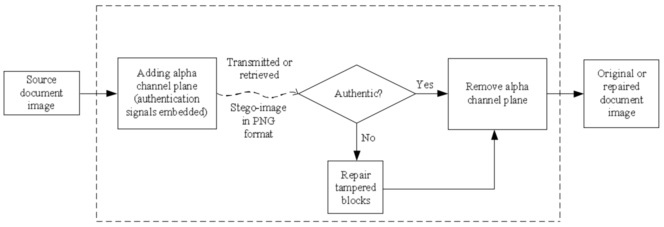

Source document image
Adding alpha channel plane (authentication signals embedded)
Transmitted or retrieved
Stego-image in PNG format
Authentic?
Yes
No
Remove alpha channel plane
Repair tampered blocks
Original or repaired document image

Fig. 6. Framework of proposed document image authentication method.

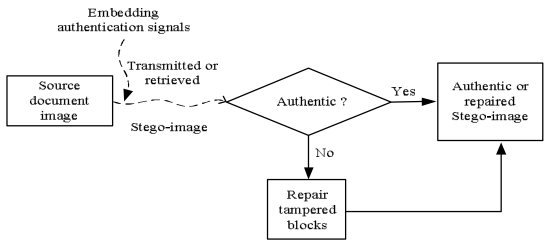


Fig. 7. Framework of a conventional image authentication method.

B. Measures for Security Enhancement

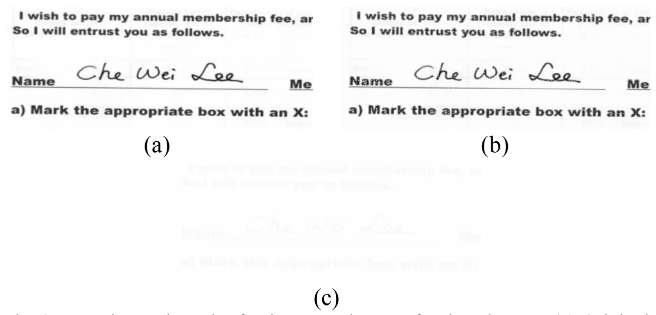
The secret key K, which is used to randomize the pixel positions for embedding the mapped partial shares q_3' through q_6' mentioned in Step 9 of Algorithm 3, provides a measure to protect the shares. More specifically, as described in Algorithm 3, each block in the alpha channel plane may be regarded to consist of two parts, *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 q_1' and q_2', and the second part for keeping the last four partial shares q_3' through q_6' 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 $m\times n$ is the size of the cover image; $m\times n/6$ is the total number of blocks, each with six pixels; and $(m\times n) - (m\times n/6)\times 2$ is the total number of pixels in the blocks other than those in the first parts of all the blocks. This probability obviously is 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 *randomization* of the constant values of x_1 through x_6 used in Step 6 of Algorithm 3 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, $(1, q_1)$ and $(2, q_2)$, can be forged easily, 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 $0 \le x_i < p$ (= 17) [11]. Then, the probability of guessing correctly all these values for all the $m\times n/6$ blocks in a stego-image can be figured out 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 $m\times n$.

V. Experimental Results and Comparison with Other Methods

A. Experimental Results Using a Document Image of a Signed Paper

The first results we show here come from our experiments using a document image of a signed paper shown in Fig. 8(a). The result of applying Algorithm 3 to embed share data into Fig. 8(a) is shown in Fig. 8(b). As can be seen, the stego-image shown in the latter is visually almost identical to the cover image shown in the former although the alpha channel content of the latter image includes the embedded data. As a comparison, Fig. 8(c) shows a result created similarly but without conducting Step 7 of Algorithm 3 which maps the original partial share values into the small interval of alpha channel values ranging from 238 through 254. An obvious opaque effect (nearly white) appears in Fig. 8(c).



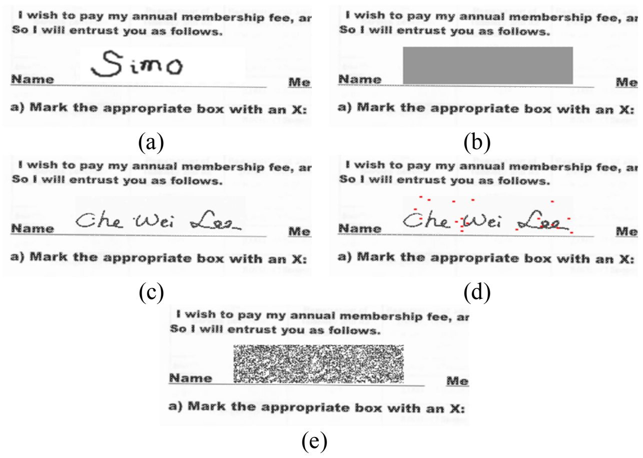
I wish to pay my annual membership fee, ar
So I will entrust you as follows.
Name Che Wei Lee Me
a) Mark the appropriate box with an X:

Fig. 8. Experimental result of a document image of a signed paper. (a) Original cover image. (b) A stego-image with embedded data. (c) Another stego-image created without conducting partial share value mapping.

We have also conducted image-modification attacks to the stego-images using two common image editing operations, namely, superimposing and painting. Tampered images yielded by the superimposing operation are presented in Figs. 7 through 10. It was observed from these experimental results that the superimposing operation, like that provided by the image editing software Adobe Photoshop or Corel PhotoImpact, destroys the content of the alpha channel values by replacing all the original alpha channel values at the attacked part with the new values of 255. Since the largest alpha channel values created by the proposed method is 254 (see Step 7 of Algorithm 3), all pixels with the unique values of 255 in the alpha channel plane may be easily detected as tampered by a modified version of Step 3 of Algorithm 4, which we describe as follows:

Step 3′ (*Check of superimposing attacks and extraction of the hidden authentication signal*) Check if both q_1' and q_2' are 255; if so, *then* (1) regard the corresponding block B' in I' as attacked by superimposing; (2) mark B', B_b', and all the partial shares embedded in B_α' as tampered; and (3) go to Step 6; *otherwise*, perform the original operations of Step 3 of Algorithm 4.

Fig. 9(a) shows the result of superimposing a white rectangular shape with a fake signature "Simo" on the genuine signature "C. W. Lee" in the stego-image of Fig. 8(b). Fig. 9(b) shows the authentication result yielded by Algorithm 4, with the gray blocks indicating the detected tampered image parts. As can be seen, the superimposing rectangular part on the signature C. W. Lee has been detected completely. For each of the detected tampered blocks, if at least two untampered shares of it can be collected, its original content can be repaired, yielding the result shown in Fig. 9(c); otherwise, the tampered block is left *unrepaired* as shown by the red dots in Fig. 9(d). Additionally, we show the data repair result obtained with a wrong key in Fig. 9(e). As can be seen, the repair work cannot be accomplished correctly; the result is just noise.



I wish to pay my annual membership fee, ar
So I will entrust you as follows.
Name Simo Me
a) Mark the appropriate box with an X:

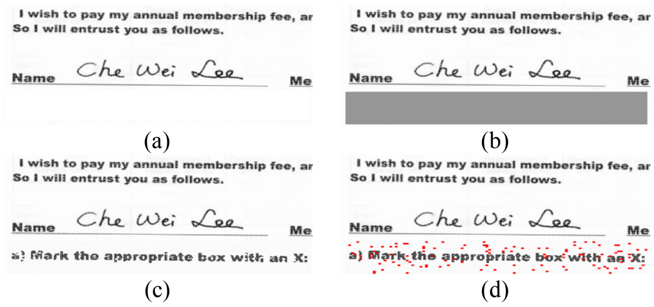
Fig. 9. Authentication result of a PNG document image of a signed paper attacked by superimposing a white rectangular shape on the signature in Fig. 6(b). (a) Tampered image yielded by the superimposing operation. (b) Result with tampered blocks detected and marked as gray. (c) Data repair result. (d) Data repair result with red dots indicating unrepaired tampered blocks. (e) Erroneous data repair result obtained with a wrong key.

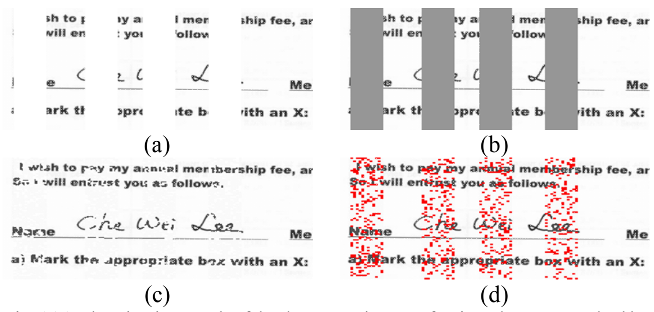
In Fig. 10(a), a text line under the signature in the signed paper disappeared after a white rectangular band was superimposed on it. The results of image authentication and repairing are shown in Figs. 10(b) and 10(c), respectively. Though some blocks are not repaired as indicated by the red dots in Fig. 10(d), the repair result of the text line is visually well recognizable. To test further the performance of the proposed method, we enlarged tampered areas during attacks and a result is shown in Fig. 11. It can be seen from the figure that the data repair result becomes worse when the tampered area grows. This is reasonable because when the tampered area becomes larger, fewer partial shares for data repairing will survive.

Table 1 includes the statistics of the performance of the proposed method shown by the above experimental results in terms of the five parameters: *tampering ratio*, *detection ratio*, *repair ratio*, *false acceptance ratio*, and *false rejection ratio*, which are defined in the following:

(1) *tampering ratio* = (the number of tampered blocks)/(the total number of blocks);
(2) *detection ratio* = (the number of detected blocks)/(the number of tampered blocks);
(3) *repair ratio* = (the number of repaired blocks)/(the number of detected blocks);
(4) *false acceptance ratio* = (the number of tampered blocks marked as untampered)/(the total number of tampered blocks);
(5) *false rejection ratio* = (the number of untampered blocks marked as tampered)/(the total number of untampered blocks).

Note that the detection ratios are all 100% due to the ease in detection of the alpha channel values of 255 (using Step 3′ described above) at image parts attacked by superimposing, as mentioned previously. Likewise, the alpha channel value corresponding to an intact block will not be 255 and can be easily checked to be so, yielding a false rejection rate of 0%. On the contrary, the alpha channel value corresponding to a tampered block is 255 which is easy to check as well, yielding a false acceptance rate of 0%.

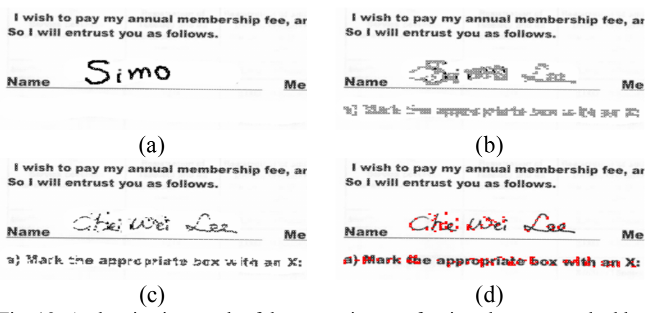

I wish to pay my annual membership fee, ar
So I will entrust you as follows.
Name
Che Wei Lee
Me
a) Mark the appropriate box with an X:
(a)
(b)
(c)
(d)
Fig. 10. Authentication result of the document image of a signed paper attacked by superimposing a white rectangular shape on a piece of text in Fig. 6(b). (a) Tampered image yielded by the superimposing operation. (b) Result with tampered blocks detected and marked as gray. (c) Data repair result. (d) Data repair result with red dots indicating unrepaired tampered blocks.


I wish to pay my annual membership fee, ar
So I will entrust you as follows.
Name
Che Wei Lee
Me
a) Mark the appropriate box with an X:
(a)
(b)
(c)
(d)
Fig. 11Authentication result of the document image of a signed paper attacked by superimposing white raster rectangular shapes on the content in Fig. 6(b). (a) Tampered image yielded by the superimposing operation. (b) Result with tampered blocks detected and marked as gray. (c) Data repair result. (d) Data repair result with red dots indicating unrepaired tampered blocks.

The content of a stego-image may be modified as well by the common operation of painting provided by well-known image editing software. Again, painting using Adobe Photoshop will replace the alpha channel values by 255, just like the superimposing operation mentioned previously. However, it was found in this study that the painting operation provided by Corel PhotoImpact does *not* change the alpha channel values. Therefore, we conducted experiments of stego-image attacks using this type of painting. Some results are given in Figs. 12 through 14. In Fig. 12, the painting operation was used to smear background gray values on the original signature “C. W. Lee” and write a fake signature “Simo” on it as shown in Fig. 12(a). Fig. 12(b) shows the authentication result in which gray blocks were used again to indicate image parts where mismatching authentication signals were detected. Note that the smeared part (the signature C. W. Lee) and the added part (the signature “Simo”) have both been revealed by the authentication process. And it is can be seen that some black parts exist within the gray region, meaning that these parts, though tampered, were not detected by the proposed method. This is due to the fact that there is actually a probability of 1/4 for an erroneous block authentication to occur because only two bits are created as the signal for block authentication (see Step 8 of Algorithm 3 or Step 5 of Algorithm 4). Though this probability seems large, yet due to the use of the secret sharing technique for generating multiple shares which are embedded randomly to survive attacks, correctly authenticated and repaired blocks still yield results with contents *visually recognizable* for image integrity judgment, as shown by Figs. 12(c) and 12(d).

Table 1. Statistics of experimental results of attacks using superimposing.

Experimental result (image size = 340×158)	No. of blocks	No. of tampered blocks (tampering ratio)	No. of detected blocks (detection ratio)	No. of repaired blocks (repair ratio)	false acceptance ratio	false rejection ratio
Exp. 1 shown in Fig. 9	8927	1488 (16.67%)	1488 (100%)	1469 (98.72%)	0%	0%
Exp. 2 shown in Fig. 10	8927	2200 (24.64%)	2200 (100%)	2087 (94.86%)	0%	0%
Exp. 3 shown in Fig. 11	8927	3792 (42.48%)	3792 (100%)	3011 (79.40%)	0%	0%


I wish to pay my annual membership fee, ar
So I will entrust you as follows.
Name
Simo
Che Wei Lee
Me
a) Mark the appropriate box with an X:
(a)
(b)
(c)
(d)
Fig. 12. Authentication result of document image of a signed paper attacked by painting white color on the original signature and texts, and replacing the signature by a fake one in Fig. 6(b). (a) Tampered image yielded by the painting operation. (b) Result with tampered blocks detected and marked as gray. (c) Data repair result (d) Data repair result with red dots indicating unrepaired blocks.

In Fig. 13, the signature was removed by replacing it with the background gray value using painting. The results of image authentication and data repairing are shown in Figs. 13(b) through 13(d). Fig. 14 shows the results of removing the entire image content by painting. In both cases, the untouched content of the alpha channel values still yields repair results with their contents recognizable to a certain degree.

It is noted here that, when a stego-image is tampered with by painting which does *not* changed the content of the alpha channel plane, the hidden authentication signals and data for repairing are not destructed. Therefore, the computed authentication signals from the alpha channel values are always true, and as long as the computed authentication signal is not identical to the extracted authentication signal for a block, the block will be marked as having been tampered with. This explains why the false rejection rate is 0%. However, as mentioned previously there is a probability of 1/4 for an erroneous block authentication to occur because only two bits are created as the signal for block authentication, and this leads to a false acceptance ratio of at most 25%. These reasonings are verified by the performance statistics of Figs. 12 through 14 listed in Table 2. The table also shows that the detection ratios are roughly around 75% as the attacked part becomes large (like the case of Fig. 14), meeting the probabilistic expectation of 1/4 block authentication misses just mentioned. Also, the false acceptance ratios are smaller than 25%, as expected.

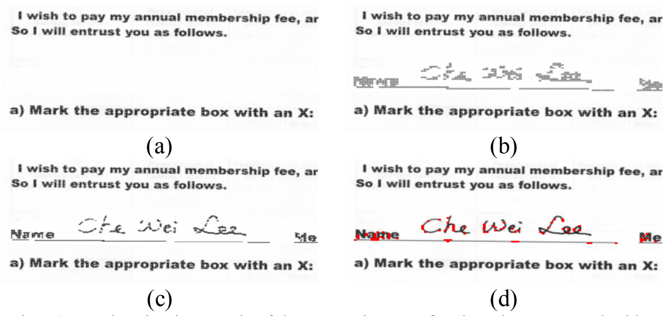


Fig. 13. Authentication result of document image of a signed paper attacked by painting white color on signature in Fig. 6(b). (a) Tampered image yielded by the painting operation. (b) Result with tampered blocks detected and marked as gray. (c) Data repair result. (d) Data repair result with red dots indicating unrepaired tampered blocks.

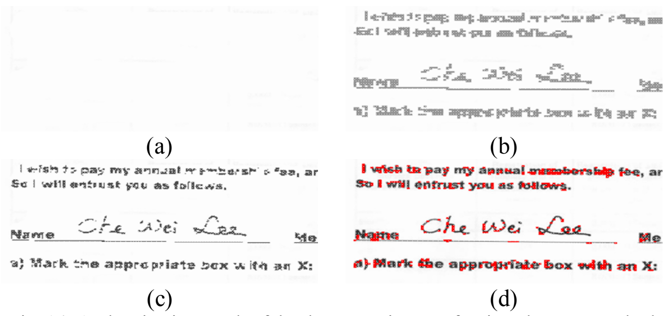


Fig. 14. Authentication result of the document image of a signed paper attacked by painting white color on entire content of Fig. 6(b). (a) Tampered image yielded by the painting operation. (b) Result with tampered blocks detected and marked as gray. (c) Data repair result. (d) Data repair result with red dots indicating unrepaired tampered blocks.

Table 2. Statistics of experimental results of attacks using painting operations.

Experimental result (image size = 340×158)	No. of blocks	No. of tampered blocks (tampering ratio)	No. of detected blocks (detection ratio)	No. of repaired blocks (repair ratio)	false accep-tance ratio	false rejection ratio
Exp. 4 shown in Fig. 12	8927	442 (5%)	396 (89.59%)	396 (100%)	10.41%	0%
Exp. 5 shown in Fig. 13	8927	781 (8.75%)	635 (81.31%)	635 (100%)	18.69%	0%
Exp. 6 shown in Fig.14	8927	1591 (17.82%)	1221 (76.7%)	1221 (100%)	23.26%	0%

B. Experimental Results Using a Document Image of a Check

Experimental results yielded by the use of a document image of a check are shown in Figs. 15(a) through 15(f) where Figs. 15(a) and 15(b) are respectively the cover document image and the stego-image generated by the proposed method. Both the amount-related text and numerals in the check image were modified as shown in Fig. 15(c). Fig. 15(d) shows that the tampering was successfully detected and marked as gray, and the result of data repairing is shown in Fig. 15(e). Finally, Fig. 15(f) shows the result of data repairing in which two unrepaired tampered blocks are shown in red. Also, we show the statistics of this experiment in Table 3.

At last, we put some noise onto the stego-image of Fig. 15(b) as an intended attack to the image, yielding the noisy image of Fig. 16(c) which then was authenticated by Algorithm 3 to get the result of Fig. 16(d) with detected tampered blocks marked in gray. The data repair result is shown in Fig. 16(e). As can be seen, some noise was not detected. The reason, as mentioned previously, is again that there is a probability of 1/4 for an erroneous block authentication to occur. Fig. 16(f) shows the unrepaired pixels in red. And the statistics of this experiment is also included into Table 3, where we can see that the false acceptance rate is about 17.12% resulting from the above-mentioned undetected noise.

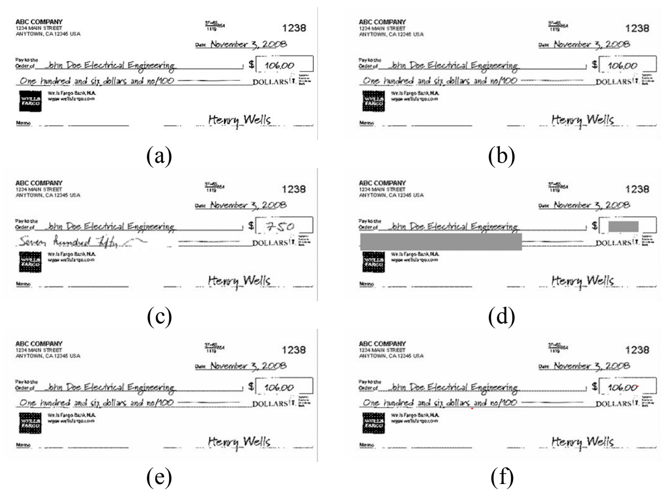


Fig. 15. Authentication result of an image of a check in PNG format attacked by superimposing counterfeit number "750" located at right side and text "Seven hundred fifty" located at left side. (a) Original cover image. (b) A stego-image with embedded data. (c) Tampered image by the superimposing operation. (d) Result with tampered blocks detected and marked as gray. (e) Data repair result. (f) Unrepaired pixels shown in red (none for this example).

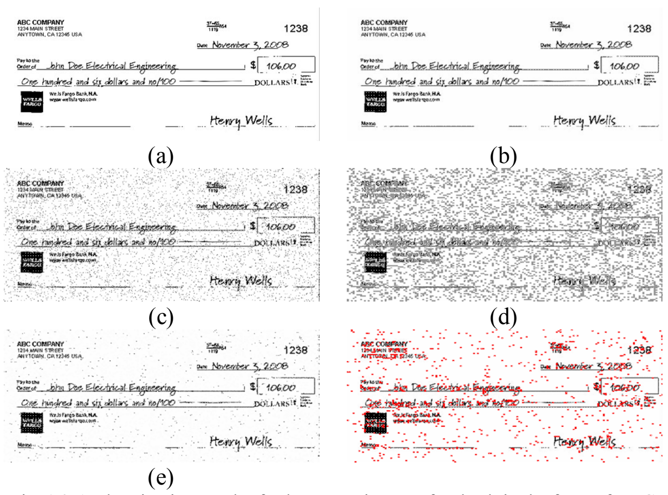


Fig. 16. Authentication result of a document image of a check in the form of PNG attacked by added noises. (a) Original cover image. (b) A stego-image with embedded data. (c) Tampered image with added noises. (d) Result with tampered blocks detected and marked as gray. (e) Data repair result. (f) Data repair result with red dots indicating unrepaired tampered blocks.

Table 3. Statistics of experimental results of using an image of check.

Experimental result (image size = 504×226)	No. of blocks	No. of tampered blocks (tampering ratio)	No. of detected blocks (detection ratio)	No. of repaired blocks (repair ratio)	false accep-tance ratio	false rejection ratio
Exp. 7 shown in Fig. 15	18984	1434 (7.55%)	1434 (100%)	1432 (99.86%)	0%	0%
Exp. 8 shown in Fig. 16	18984	6695 (35.27%)	5549 (82%)	5549 (100%)	17.12%	0%

C. Comparison of Performances with Other Methods

A comparison of the capabilities of the proposed method with those of four existing methods is shown in Table 4. All but the proposed method will create distortion in the stego-image during the authentication process. More importantly, only the proposed method has the capability of repairing the tampered parts of an authenticated image.

Furthermore, among the methods with tampering localization capabilities at the block level like Yang and Kot [5], Tzeng and Tsai [8], and the proposed method, the proposed method provides a finer authentication precision with the block size of 2×3. Specifically, the method in [5] needs larger *macro-blocks* to yield pixel flippabilities for embedding authentication data. In the case of using smaller blocks, Tzeng and Tsai's method [8] has a high possibility to generate noise pixels as mentioned in [6], and so they conducted experimental results with the larger block size of 64×64.

Table 4. Comparison of document image authentication methods.

	Distortion in stego-image	Tampering localization capability	Repair capability	Reported authentication precision	distribution of authenticated image parts	Manipulation of data embedding
Wu & Liu [4]	Yes	No	No	Macro-block	Non-blank part	Pixel flippability
Yang & Kot [5]	Yes	Yes	No	33×33 block	Non-blank part	Pixel flippability
Yang & Kot [6]	Yes	No	No	Macro-block	Non-blank part	Pixel flippability
Tzeng & Tsai [8]	Yes	Yes	No	64×64 block	Entire image	Pixel replacement
Proposed method	No	Yes	Yes	2×3 block	Entire image	Alpha channel pixel replacement

As to the distribution of *authenticated image parts*, because there exists no flippable pixel for use by the methods of [4-6] to embed data in *all-white* regions (such as marginal regions) of a document image, the distribution of authenticated image parts tends to be restricted to be on lines or strokes in the document, while the proposed method does not have this limitation. Nevertheless, in [4-6] the authenticity of an image part including such all-white regions can be still ensured by the use of cryptographic signatures embedded in other regions of the image. At last, the methods of [4-6] manipulate pixel flippability and the method of [8] enforces pixel replacement for the aim of data embedding. The proposed method is the only one which makes use of the alpha channel plane instead of the bit plane.

VI. Conclusions

A new blind image authentication method with a data repair capability for binary-like grayscale document images based on secret sharing has been proposed. Both the generated authentication signal and the content of a block are transformed into partial shares by the Shamir method, which are then distributed in a well-designed manner into an alpha channel plane to create a stego-image in the PNG format. The undesired opaque effect visible in the stego-image coming from embedding the partial shares is eliminated by mapping the share values into a small range of alpha channel values near their maximum transparency value of 255.

In the process of image block authentication, a block in the stego-image is regarded as having been tampered with if the computed authentication signal does not match that extracted from corresponding partial shares in the alpha channel plane. For self-repairing of the content of a tampered block, the reverse Shamir scheme is 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 were also proposed. Experimental results have been shown to prove the effectiveness of the proposed method. Future studies may be directed to choices of other block sizes and related parameters (prime number, coefficients for secret sharing, number of authentication signal bits, etc.) to improve data repair effects. Applications of the proposed method to authentication and repairing of attacked color images may also be tried.

References

[1] C. S. Lu and H. Y. M. Liao, "Multipurpose watermarking for image authentication and protection," *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1579–1592, Oct. 2001..

[2] M. U. Celik, G. Sharma, E. Saber and A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization," *IEEE Trans.on Image Processing*, vol. 11, no. 6, pp. 585–595, June 2002.

[3] Z. M. Lu, D. G. Xu and S. H. Sun, "Multipurpose image watermarking algorithm based on multistage vector quantization" *IEEE Trans. on Image Proc.*, vol. 14, pp. 822–831, June 2005.

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

[5] H. Yang and A. C. Kot, "Binary image authentication with tampering localization by embedding cryptographic signature and block identifier," *IEEE Signal Processing Letters*, vol. 13, no. 12, pp. 741–744, Dec. 2006.

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

[7] H. Y. Kim and A. Afif, "Secure authentication watermarking for halftone and binary images," *Int'l Journal of Imaging Systems and Technology*, vol. 14, no. 4, pp. 147–152, 2004.

[8] 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 Communications Letters*, vol. 7, no. 9, pp. 443–445.

[9] Y. Lee, J. Hur, H. Kim, Y. Park and H. Yoon, "A new binary image authentication scheme with small distortion and low false negative rates," *IEICE Trans. on Communications*, vol. E90-B, no. 11, Nov. 2007.

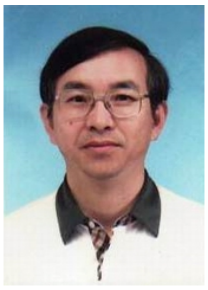
[10] Y. Lee, H. Kim, and Y. Park, "A new data hiding scheme for binary image authentication with small image distortion," *Information Sci.*, vol. 179, no. 22, pp. 3866–3884, Nov. 2009.

[11] A. Shamir, "How to share a secret," *Communication of ACM*, vol. 22, pp. 612–613, 1979.

[12] C. C. Lin and W. H. Tsai, "Secret image sharing with steganography and authentication," *Journal of Systems & Software*, vol. 73, pp. 405–414, 2004.

[13] W. H. Tsai, "Moment-preserving thresholding: a new approach," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 377-393, 1985.

Che-Wei Lee receives the B. S. degree in civil engineering and the M. S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2002 and 2005, respectively. He is a Ph. D. student in the Department of Computer Science at National Chiao Tung University since 2005. His research interests include information hiding, image processing, and video technologies.



Wen-Hsiang Tsai received the B.S. degree in EE from National Taiwan University, Taiwan, in 1973, the M.S. degree in EE from Brown University, USA in 1977, and the Ph.D. degree in EE from Purdue University, USA in 1979. Since 1979, he has been with National Chiao Tung University (NCTU), Taiwan, where he is now a Chair Professor of Computer Science. At NCTU, he has served as the Head of the Dept. of Computer Science, the Dean of General Affairs, the Dean of Academic Affairs, and a Vice President. From 1999 to 2000, he was the Chair of the Chinese Image Processing and Pattern Recognition Society of Taiwan, and from 2004 to 2008, the Chair of the Computer Society of the IEEE Taipei Section in Taiwan. From 2004 to 2007, he was the President of Asia University, Taiwan.

Dr. Tsai has been an Editor or the Editor-in-Chief of several international journals, including *Pattern Recognition*, the *International Journal of Pattern Recognition and Artificial Intelligence*, and the *Journal of Information Science and Engineering*. He has published 146 journal papers and 228 conference papers received many awards, including the Annual Paper Award from the Pattern Recognition Society of the USA; the Academic Award of the Ministry of Education, Taiwan; the Outstanding Research Award of the National Science Council, Taiwan; the ISI Citation Classic Award from Thomson Scientific, and more than 40 other academic paper awards from various academic societies. His current research interests include computer vision, information security, video surveillance, and autonomous vehicle applications. He is a Life Member of the Chinese Pattern Recognition and Image Processing Society, Taiwan and a Senior Member of the IEEE.