Authentication of Binary Document Images in PNG Format Based on A Secret Sharing Technique*

Che-Wei Lee
Institute of Multimedia Engineering
National Chiao Tung University, Hsinchu, Taiwan
lcw.cs94g@nctu.edu.tw

☆Wen-Hsiang Tsai
Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan
whtsai@cis.nctu.edu.tw

***Abstract*—A new authentication method utilizing the secret sharing technique to resist attacks with a data recovery capability for binary document images in PNG format is proposed. An authentication signal is generated for each block of a cover image, which together with the block content is transformed into several shares using the Shamir secret sharing scheme. The shares then are embedded into the alpha channel plane of the input PNG image in a carefully designed manner. 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 recovery is conducted for each tampered block by a reverse Shamir scheme after collecting enough shares from unmarked blocks. Experimental results prove the effectiveness of the proposed method for real applications.**

***Keywords* — image authentication, binary document image, secret sharing, data hiding, PNG image.**

I. INTRODUCTION

For the purpose of preserving important documents, transforming them into digital form, such as digital images, is a way to satisfy the requirement. As a result of the nature of digitized data, it is not difficult to tamper with the content of a digital image imperceptibly. Therefore, designing effective methods for document image authentication [1]–[3] to ensure the integrity and authenticity of digital document images is desirable. It is also hoped that if a critical image part is authenticated to have been altered illicitly, its original content can be recovered. Such image content verification and self-recovery capabilities are useful for authentication of many kinds of digital documents, such as signed forms, secret documents, scanned checks, important certificates, circuit diagrams, art drawings, design drafts, last will and testament documents, and so on.

*This work was supported financially by the Ministry of Economic Affairs under Projects Nos. MOEA 97-EC-17-A-02-S1-032 and MOEA 98-EC-17-A-02-S2-0047 in the Technology Development Programs for Academia.

☆Also with Department of Information Communication, Asia University, Taiwan.

In particular, the content authentication problem is difficult for the binary document image because of its simple binary nature. Nevertheless, several image authentication methods for binary images have been proposed in the past. Tzeng and Tsai [4] proposed a binary image authentication method by embedding randomly-generated authentication codes into image blocks and designed a code holder to reduce the image distortion resulting from data embedding. Wu et al. [5] manipulated the so-called flippable pixels, which yield less noticeable results when they are flipped in a block pattern, to enforce specific relationships to embed data for authentication and annotation of binary images. Yang et al. [6] proposed a pattern-based data hiding method for binary image authentication, which assesses the flippability of a pixel more efficiently by a connectivity-preserving criterion to generate a watermarked image with low distortion.

In this study, a method for binary PNG image authentication with an additional self-recovery capability for repairing possibly attacked image data is proposed. The method is based on the so-called (k, n)-threshold scheme proposed by Shamir [7] for secret sharing in which a secret message is transformed into n *shares* which are then kept by n participants; and when k of the shares, not necessarily all of them, are collected, the secret message can be recovered.

The proposed method uses the secret sharing scheme to "carry" authentication signals and image content data via the use of the shares to enhance the robustness of the proposed method in handling possible attacks to the embedded data. And the alpha channel of a binary PNG image is utilized to accommodate these shares. After that, the resulting stego-image with perceptible white noise caused by these shares is further removed by mapping the computed share values into suitable ranges.

For image content recovery, it is always desirable to recover a tampered block *losslessly* if the block is authenticated to have been attacked. The proposed approach here is to distribute randomly a multiple of the original image data in the form of shares into the alpha channel. Such randomness will allow the data to survive attacks without being totally destroyed, thus

promoting the security of the embedded data. Two schemes for such content self-recovery are developed for dealing with image alterations made by painting and superimposion operations, respectively, which are available in most image processing software packages.

The remainder of this paper is organized as follows. In Section 2, the Shamir method for secret sharing is reviewed first. In Section 3, the details of the proposed method are described. Some experimental results are shown in Section 4. Finally, conclusions are made in Section 5.

II. Review of Shamir's Method for Secret Sharing

In the (k, n)-threshold secret sharing method proposed by Shamir [7], a secret d in the form of an integer is transformed into shares which are kept by n participants, and as long as k of the n shares are collected, the original secret can be recovered accordingly, where $k \le n$. 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 not larger than n.

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

1. Choose randomly a prime number p which is larger than d.
2. Select $k-1$ integer values $c_1, c_2, \ldots, c_{k-1}$ within the range of 0 through $p-1$.
3. Select n distinct real values $x_1, x_2, \ldots, x_n$.
4. Use the following $(k-1)$-degree polynomial to compute n function values $F(x_i)$:

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

where $i = 1, 2, \ldots, n$.

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, including 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. This explains the term, *threshold*, for k and the name, (k, n)-*threshold*, for the Shamir method [7]. Below is a description of the mentioned equation solving for secret recovery.

Algorithm 2:* secret *recovery.

Input: m shares in the form of $(x_j, F(x_j))$ collected from the n participants where $1 \le j \le n$, $k \le m \le n$, and k is the threshold mentioned in Algorithm 1.

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

Steps*:*

1. Collect any k of the m shares, say, $(x_{i_1}, F(x_{i_1}))$, $(x_{i_2}, F(x_{i_2}))$, …, $(x_{i_k}, F(x_{i_k}))$ and use them to set up the following equations:

$$F(x_{i_j}) = (d + c_1 x_{i_j} + c_2 x_{i_j}^2 + \ldots + c_{k-1} x_{i_j}^{k-1})_{\bmod p}, \qquad (2)$$

where $j = 1, 2, \ldots, k$ and $1 \le i_j \le n$.

2. Solve the k equations above by Lagrange's interpolation to obtain the desired secret value d [8] as follows:

$$d = (-1)^{k-1}[F(x_{i_1})\frac{x_{i_2}x_{i_3}\ldots x_{i_k}}{(x_{i_1}-x_{i_2})(x_{i_1}-x_{i_3})\ldots(x_{i_1}-x_{i_k})} + F(x_{i_2})\frac{x_{i_1}x_{i_3}\ldots x_{i_k}}{(x_{i_2}-x_{i_1})(x_{i_2}-x_{i_3})\ldots(x_{i_2}-x_{i_k})} + \ldots + F(x_{i_k})\frac{x_{i_1}x_{i_2}\ldots x_{i_{k-1}}}{(x_{i_k}-x_{i_1})(x_{i_k}-x_{i_2})\ldots(x_{i_k}-x_{i_{k-1}})}]_{\bmod p}.$$

3. Compute the values c_1 through c_{k-1} by expanding the following equality and compare the result with (2) in Step 1:

$$F(x) = [F(x_{i_1})\frac{(x-x_{i_2})(x-x_{i_3})\ldots(x-x_{i_k})}{(x_{i_1}-x_{i_2})(x_{i_1}-x_{i_3})\ldots(x_{i_1}-x_{i_k})} + F(x_{i_2})\frac{(x-x_{i_1})(x-x_{i_3})\ldots(x-x_{i_k})}{(x_{i_2}-x_{i_1})(x_{i_2}-x_{i_3})\ldots(x_{i_2}-x_{i_k})} + \ldots + F(x_{i_k})\frac{(x-x_{i_1})(x-x_{i_2})\ldots(x-x_{i_{k-1}})}{(x_{i_k}-x_{i_1})(x_{i_k}-x_{i_2})\ldots(x_{i_k}-x_{i_{k-1}})}]_{\bmod p}.$$

Step 3 in the above algorithm is included additionally for the reason that we want to compute the values of c_i in the proposed method. In other applications, if only the secret need be recovered, this step may be omitted.

III. Authentication and Tampered Data Recovery

In the proposed method, given an input *cover image* of the binary PNG format, it first divided into nonoverlapping blocks of size 2×3, and an authentication signal is generated from each of them. Next, the generated authentication signal and the block content *for possible later recovery* are gathered to form a secret message. The message is then transformed into n shares by the Shamir's (k, n)-threshold secret sharing method described by Algorithm 1. Finally, the shares are embedded into the alpha-channel plane of the cover image to form a stego-image. Later, if some, but not more than k, of the shares are destroyed in an attack, then k of the remaining untouched shares can be retrieved to recover the original image data. In this way, the chance of data survival against attacks will be raised.

A detailed algorithm for describing the above mentioned is presented in the following

Algorithm 3: authentication signal generation and embedding of shares.

Input: a binary PNG cover image I and a key K.

Output: a stego-image I' with the message data (including the authentication signals and the image data for recovery) embedded.

Steps.

Step 1. (*Beginning of looping*) In a raster-scan order, take from I a 2×3 block B with pixels p_1 through p_6.

Step 2. (*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.

Step 3. (*Creation of secret and coefficient values for secret sharing using Algorithm 1*) Create two binary values

$m_1 = a_1a_2p_1p_2$; $m_2 = p_3p_4p_5p_6$, and transform them into decimal numbers m_1' and m_2', respectively.

Step 4. (*Partial share generation*) Set p, c_i, and x_i in Eq. (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:

$$\begin{aligned} q_1 &= F(x_1) = (d + c_1x_1)_{\bmod p}, \\ q_2 &= F(x_2) = (d + c_1x_2)_{\bmod p}, \\ &\ldots, \\ q_6 &= F(x_6) = (d + c_1x_6)_{\bmod p}. \end{aligned} \quad (3)$$

Step 5. (*Mapping of partial share values*) Add 239 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 239 through 255.

Step 6. (*Embedding of two partial shares at the current block*) Take the block B' in the alpha-channel plane L which in position corresponds to B in I, select in the raster-scan order the first two pixels in B', and set their values to be q_1' and q_2', respectively.

Step 7. (*Embedding the remaining partial shares at random positions*) Use the input key K to select randomly four pixels in L *outside* B' and *unselected yet* in this step, and set their values respectively to be the four mapped partial shares q_3' through q_6' of B generated in Step 5 above.

Step 8. (*End of looping*) If there exists any unprocessed block in I, then go to Step 1; otherwise, take the final I as the desired stego-image I'.

The possible values of q_1 through q_6 yielded by the formulas in (3) above and inserted in the alpha channel plane of I are between 0 and 16 because the value of p used in (3) is 17. And after performing Step 5 of the above algorithm, the values of q_1 through q_6 become q_1' through q_6', respectively, which fall in a small range of integers from 239 to 255, resulting in a *nearly uniformly transparent binary* PNG image with a steganographic effect.

Now, the processes of verification of a stego-image and self-recovery of the original image content are described in the following.

Algorithm 4: secret data extraction from a stego-image.

Input: a stego-image I' generated by Algorithm 3 and the key K used there.

Output: an image I_r with tampered blocks being marked and their data being recovered losslessly.

Steps.

Step 1. (*Beginning of looping*) Take in a raster-scan order from I' a block B with pixels p_1 through p_6 and let their alpha-channel values be denoted as q_1' through q_6', respectively.

Step 2. (*Extraction of hidden authentication signals*) Perform the following steps to extract the embedded authentication signals in the alpha channel plane L.

(1) Take the block B' in L, which in position corresponds to B.
(2) Take in a raster-scan order the first two pixels in B', and let their values be denoted as q_1' and q_2', respectively.
(3) Subtract 239 from q_1' and q_2' to obtain q_1 and q_2, respectively.
(4) Take q_1 and q_2 as *two partial shares of* B'; and with them as input, perform Algorithm 2 to extract the two values d and c_1 (the secret and the first coefficient value) from q_1 and q_2 as output.
(5) Transform d and c_1 into 4-bit numbers, concatenate them in order to form an 8-bit string S, and take the first two bits of S as a_1 and a_2.

Step 3. (*Computation of authentication signals*) For each block B in I' with pixel values p_1 through p_6, compute for it two authentication signals a_1' and a_2' as follows:

$$a_1' = p_1 \oplus p_2 \oplus p_3;\ a_2' = p_4 \oplus p_5 \oplus p_6.$$

Step 4. (*Matching of hidden and computed authentication signals and marking tampered blocks*) Match respectively the authentication signals a_1 and a_2 just extracted with the signals $a_1'$and a_2' computed previously, and if any mismatch occurs, mark B and all the mapped partial shares embedded in B' as *tampered*.

Step 5. (*End of looping*) If there exists any unprocessed block in I', then go to Step 1; otherwise, continue.

Step 6. (*Extraction of the remaining partial shares*) For each block B' in the alpha-channel plane L, perform the following steps to extract the remaining four partial shares of B' from the alpha channel values of the other blocks.

(1) Use K to collect the four same pixels which were randomly selected for B' in Step 7 of Algorithm 3, and take out the data q_3', q_4', q_5', and q_6' embedded in them.
(2) Subtract 239 from each of q_3' through q_6', respectively, to obtain four partial shares q_3 through q_6.

Step 7. (*Repair the tampered regions*) For each block B' marked as tampered, perform the following steps to recover its original content.

(1) From the six previously computed or collected shares q_1 through q_6 of B', choose, if possible, two shares, denoted as r_1' and r_2', which are *not* marked as tampered.
(2) Subtract 239 from r_1' and r_2' to obtain r_1 and r_2, respectively.

(3) Take r_1 and r_2 as *two partial shares of B′*; and with them as input, perform Algorithm 2 to extract the values of d and c_1 (the secret and the first coefficient value) from r_1 and r_2 as output.
(4) Transform d and c_1 into 4-bit numbers and concatenate them in order to form an 8-bit string S'.
(5) Take the last six bits from S' to replace the pixel values of the block B of the intensity channel corresponding to B' as the result of data recovery.

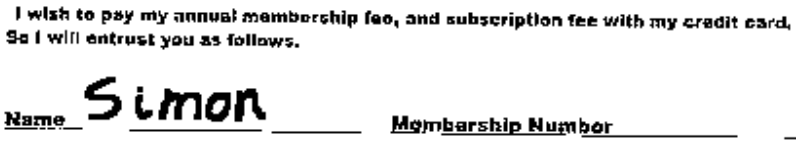
Step 8. Take the final I' as the desired self-recovered image I_r.

Common image superimposing operations used by application packages like Photoshop will change the alpha channel values of a PNG binary image. This leads to the loss of the authentication signals and the embedded data for recovery. Therefore, partial shares embedded in the alpha-channel values of a block which is attacked by such operations will become useless and should not be used further for tampered block recovery, as performed by Algorithm 4 above (see Step 7(1)).

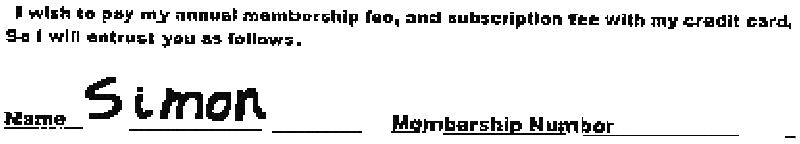
IV. Experimental results

To test the proposed method, some experiments have been conducted. We used a signed paper in the experiments and scanned it to generate a digital file in the form of a binary PNG image as shown in Fig. 1(a). The result of applying the proposed method using Algorithm 3 to embed authentication signals into Fig. 1(a) is shown in Fig. 1(b). We then used the painting operation to smear white color on the signature "Simon" and write a fake one "Mac" to replace it, yielding an image as shown in Fig. 1(c). Fig. 1(d) shows the authentication result with Fig. 1(c) as input, in which we use *gray* blocks to indicate altered image parts where mismatching authentication signals were detected. Note that the smeared parts (the signature Simon) and the added parts (the signature Mac) are both revealed. And it is can be seen that some black parts exist within the gray region, meaning that these parts, though tampered with, are not detected by the method. These misses result from a chance of matching when a computed authentication signal from the altered parts is identical in value to the corresponding signal extracted from the partial shares embedded in the alpha channel. At last, the result of data recovery is shown in Fig. 1(e).

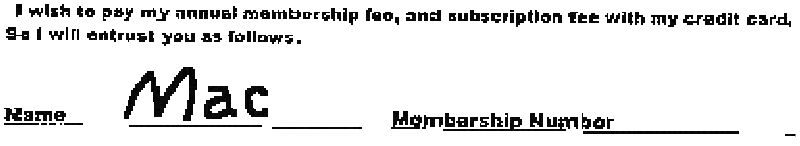
For the case of tampered images resulting from an attack of collage, Fig. 1(f) shows the result of superimposing a white rectangular shape to replace the signature "Simon" in the stego-image of Fig. 1(b). Fig. 1(g) shows the authentication result, also with gray blocks indicating detected altered image parts. As can be seen, the collaged rectangular part is successfully detected by the proposed method. This comes from the fact that the alpha channel values of a tampered block resulting from image superimposing will change, leading to destruction of the embedded shares in the alpha channel plane. Nevertheless, for each of such tampered blocks, only when two untampered shares of it are collected can its original content be recovered as shown in Fig. 1(h).



(a)



(b)



(c)

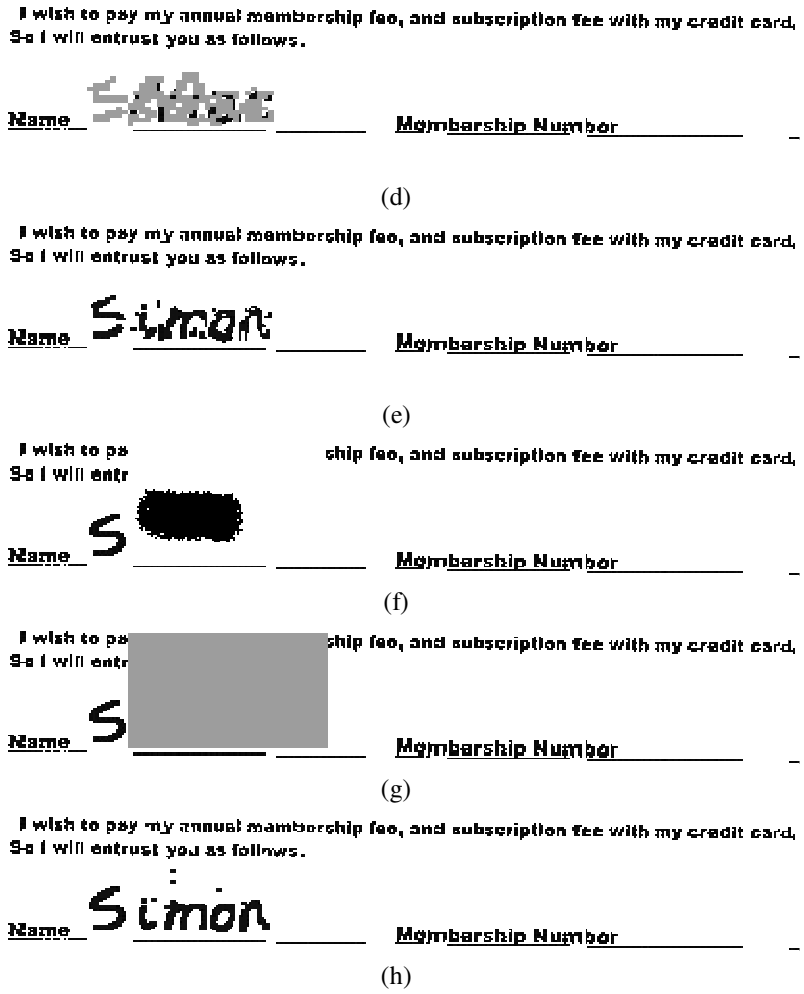


Fig. 1. Authentication result of a binary document image in PNG format tampered with by painting and superimposing. (a) Original cover image. (b) A stego-image. (c) A tampered image yielded by painting a false signature on (b). (d) Result with tampered blocks detected and marked as gray. (f) A tampered image yielded by superimposing a white rectangular shape on (b). (g) Result of authentication with tampered blocks detected and marked as gray. (h) Result of data recovery with some unrepaired tampered blocks also shown.

V. Conclusions

A new image authentication method with a data recovery capability for binary PNG images based on secret sharing has been proposed. Both the generated authentication signal and the original content of a block are transformed into shares by the Shamir method, which are then distributed in a carefully-designed manner into the alpha-channel plane. In the process of image authentication, a block is regarded as tampered if there exists a mismatch between the computed authentication signal and that extracted from corresponding shares. For self-recovery of the content of a tampered block, the reverse Shamir's scheme is used to compute the original content of the block from two shares collected from untampered blocks. Experimental results showed the effectiveness of the proposed method.

References

[1] C. S. Lu and H. Y. M. Liao, "Multipurpose watermarking for image authentication and protection," *IEEE Trans. Image Process.*, vol. 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. Image Process.*, vol. 11, no. 6, pp. 585–595, Jun. 2002.

[3] Z. M. Lu, D. G. Xu and S. H. Sun, "Multipurpose image watermarking algorithm based on multistage vector quantization" *IEEE Trans. Image Process.*, vol. 14, no. 6, pp. 822–831, Jun. 2005.

[4] 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.

[5] 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.

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

[8] C. C. Lin and W. H. Tsai, "Secret image sharing with steganography and authentication," *J. of Systems and Software*, vol. 73, pp. 405-414, 2004.