A NEW PUBLIC-KEY AUTHENTICATION WATERMARKING FOR BINARY DOCUMENT IMAGES RESISTANT TO PARITY ATTACKS

Hae Yong Kim

Universidade de São Paulo, Escola Politécnica, Brazil. E-mail: hae@lps.usp.br.

ABSTRACT

An authentication watermarking technique (AWT) inserts some hidden data into an image in order to detect any accidental or malicious image alteration. Recent papers have proposed some cryptography-based AWTs for binary and halftone images, but none of them seems to be well-suited for public-key authentication of binary document images. Many of the previous public-key AWTs for binary images can be assaulted by "parity attacks." This paper proposes a new AWT for binary document images that can detect *any* modification of the watermarked document, even a single pixel flipping. Moreover, both its secret- and public-key versions are completely immune against parity attacks. Document images watermarked by the proposed technique present high visual quality, without salt-and-pepper noise.

1. INTRODUCTION

In modern digital watermarking, a data-hiding scheme means a technique to embed a sequence of bits in a host image with small visual deterioration and the means to extract it afterwards. A watermarking technique makes use of a data-hiding scheme to insert some information in the host image, in order to make an assertion about the image in the future. Watermarking techniques can be classified as either "robust" or "fragile."

Robust watermarks are typically used for copyright assertion and fingerprinting. They should not be easily removed and must resist common image manipulation procedures such as scaling, cropping, lossy compression, etc.

Fragile watermarks (or authentication watermarks) are easily corrupted by any image-processing procedure. However, watermarks for checking the image integrity and authenticity can be fragile because, if the watermark is removed, the watermark detection algorithm will correctly report the corruption of the image.

An authentication watermarking technique (AWT) can use either secret- or public-key ciphers. In a secret-key AWT, the same secret-key is used in both watermark insertion and verification. In a public-key AWT, only the owner of the private-key can insert the valid watermark, and anyone can verify the authenticity and integrity of the watermarked image using the corresponding public-key.

In the literature, there are many AWTs for continuous-tone images (e.g., [1–5]). Also, there are many data-hiding schemes for binary and halftone images (e.g., [6–8]). However, only recently some secure AWTs for binary and halftone images have been proposed [9–11]. We mean by "secure AWT" a scheme that has two properties: (1) it must detect *any* visually significant image alteration (both accidental and malicious); (2) its security must not lie on the secrecy of the algorithm but only on the secrecy of the key. Hence, a secure AWT usually relies upon cryptography.

A cryptography-based AWT for dispersed-dot halftone images named AWST (Authentication Watermarking by Self Toggling) has been recently proposed [9]. It can be used with secret- or public-key ciphers. However, when this technique is applied to binary document images, visible salt-and-pepper noise appears.

Another cryptography-based AWT for binary document images named AWTR (Authentication Watermarking by Template Ranking) has also been proposed [10, 11]. Document images watermarked by AWTR present excellent visual quality. The secret-key AWTR is secure. However, the public-key AWTR cannot securely authenticate "small" images due to the watermark adulterating technique called "parity attack." [1]

This paper proposes a new AWT for binary document images, named AWTC (Authentication Watermarking by Template ranking with symmetrical Central pixel). It is completely immune to parity attacks and consequently can authenticate even "small" images using either secret- or public-key ciphers. Images marked by AWTC do not present visible salt-and-pepper noise. This technique can detect *any* image alteration, even a single pixel flipping.

We did not apply any perceptual distortion measure to quantify the quality of watermarked images, because this analysis is beyond the scope of this paper. However, the

This work was supported in part by FAPESP under grant 2003/13752-9 and by CNPq under grants 305065/2003-3 and 475155/2004-1.

[1] The meaning of "small" depends on the length of the adopted digital signature and on the desired level of security. As a rule of thumb, only binary images with one million or more pixels can be securely authenticated by the public-key AWTR.

0-7803-9134-9/05/$20.00 ©2005 IEEE

template ranking used in AWTC can be adapted to minimize the distortion according to a specific perceptual model.

2. PREVIOUS TECHNIQUES

As we said above, there are many data-hiding schemes for binary images in the literature. A data-hiding scheme can be transformed into an AWT by dividing the host image Z in two regions Z_1 and Z_2, computing the authentication signature (AS) of Z_2, and inserting the AS into Z_1. An AS contains information about the image content that may be checked to verify its integrity. In cryptography, an AS is called message authentication code (MAC) using a secret-key cipher or digital signature (DS) using a public-key cipher.

However, some caution must be taken in transforming a data-hiding scheme into an AWT, because although the region Z_2 is well protected (with the security assured by the cryptography theory), the region Z_1 is not. For example, let us take the data-hiding scheme that inserts one bit per connected component, forcing it to have even or odd number of black pixels. A connected component can be forced to have the desired parity by toggling one of its boundary pixels. This scheme can be transformed into an AWT using the idea described above. Yet, a malicious hacker can arbitrarily alter the region Z_1, without being noticed by the AWT, as long as the parities of all connected components remain unaltered. For example, a character "a" in Z_1 region can be changed into an "e" (or any other character that contains only one connected component) as long as its parity remains unchanged. We refer to this as a "parity attack."

Recently, the data-hiding scheme named DHTR (Data Hiding by Template Ranking) [6, 12] was transformed into an AWT named AWTR (Authentication Watermarking by Template Ranking) [10, 11] by using the idea explained in the beginning of this section. DHTR consists of:

1. Divide the image into small blocks (e.g., 8×8).
2. Analyze the neighborhood (usually 3×3) of each pixel to rate its visual significance. Figure 1 shows a template ranking suggested in [11].
3. Insert one bit in each block by forcing it to have even or odd number of black pixels. If the block already has the desired parity, no change is made. Otherwise, the pixel with the lowest visual significance is flipped.
4. As different blocks may have different quantities of low-visibility pixels, it is suggested to shuffle the image before embedding the data.

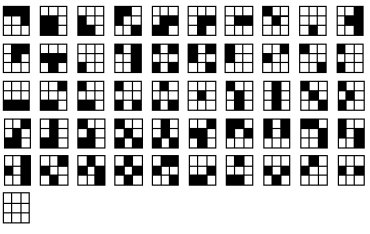


Fig. 1. A template ranking suggested to be used with AWTR [11], in increasing visual impact order. Mirrors, rotations and reverses of each pattern have the same score.

AWTR cannot detect alterations that flips an even number of pixels of a block. Hence, the parity attack against AWTR consists in modifying an even number of pixels of each Z_1 block. Kim and Queiroz have proposed two ideas to thwart parity attacks for the secret-key AWTR [10, 11]. The first idea uses the secret-key as the seed of the the pseudo-random shuffling. The second idea inserts one bit of the MAC in a block, then computes a new MAC feeding the MAC-function with the previous MAC and the modified block. Unfortunately, none of these two ideas can protect the public-key AWTR, because anyone must be abe to completely extract the DS (before decrypting it with the public-key).

There is another AWT called AWST (Authentication Watermarking by Self Toggling) [9], especially suited for dispersed-dot halftone images. AWST makes use of the data-hiding scheme named DHST (data hiding by self toggling) [7]. AWST consists of choosing a pseudo-random sequence v of pixels to bear the data, clearing them, computing the AS of the now-cleared image, and inserting the resulting AS into the pixels of sequence v. In the public-key AWST, the seed of the pseudo-random number generator must be a publicly known number. In the secret-key AWST, the seed of the generator can be either a publicly known number or the secret-key. Surprisingly, AWST cannot be assaulted by parity attacks. Consequently, both secret- and public-key versions of AWST are secure. However, an AWST-watermarked binary document image presents visible salt-and-pepper noise, because this technique does not take into account any visual impact measure.

Why do parity attacks not apply to AWST? Because the number of data-bearing Z_1 pixels is exactly equal to the length of the adopted AS. All image pixels (except the n pixels that will bear the n bits of the AS) are taken into account to compute the AS. Consequently, *any* alteration of Z_2 region can be detected because it changes the AS of the watermarked image, and *any* alteration of Z_1 region can also be detected because it changes the stored AS.

3. THE PROPOSED DATA-HIDING TECHNIQUE

The aim of this paper is to design an AWT with AWTR's visual quality and AWST's resistance to parity attacks. We

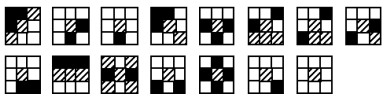


Fig. 2. A 3×3 template ranking with symmetrical central pixels in increasing impact order. Hatched pixels match either black or white pixels (note that all patterns have hatched central pixels). The score of a given pattern is that of the matching template with the lowest impact. Mirrors, rotations and reverses of each pattern have the same score.

recall that AWST chooses pseudo-randomly a sequence v of pixels in the image to bear the data. If the sequence v is selected according to some visual impact score, instead of pseudo-randomly chosen, the visual quality of the watermarked image would improve, while the AWT would remain resistant to parity attacks. However, the data embedding process usually modifies the visual scores of pixels, and consequently it is impossible to reconstruct, in the watermark verification, the same sequence v used in the watermark insertion. Fortunately, we worked out a data-hiding technique, named DHTC (Data Hiding by Template ranking with symmetrical Central pixels), that keeps the visual scores of flippable pixels unaltered with the data embedding:

1. The template ranking used by DHTC must assign the same visual impact score to the patterns that differ only by the colors of their central pixels. Figure 2 depicts a 3×3 template ranking with symmetrical central pixels. Note that all patterns have hatched central pixels. To simplify the explanation, let us assume that 3×3 patterns are used, although larger patterns may be used.

2. Divide the binary image Z to be watermarked in a sequence v of non-overlapping 3×3 pieces of image Z. The simplest of such sequence is the division of Z into regular 3×3 pieces (incomplete pieces at image borders are discarded), scanned in raster sequence (figure 3). However, this sequence will embed the data preferentially in the upper part of the image. Hence, a pseudo-random scanning (using a publicly known seed) is preferable.

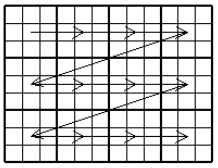


Fig. 3. A 9×12 image divided into regular 3×3 pieces and scanned in raster order.

3. Sort the sequence v in increasing order using the visual score as the primary-key and the piece's order in the original sequence v as the secondary-key.

4. Embed n bits of the data by flipping (if necessary) the central pixels of the first n pieces of the sorted v.

In the data extraction step, the sequence v of non-overlapping 3×3 pieces is constructed again. This sequence is sorted as in the insertion, resulting in the exactly same sequence v used in the insertion. Then, the values of the n first central pixels are the hidden data.

4. THE PROPOSED AWT

DHTC can be easily transformed into a secure AWT, named AWTC (Authentication Watermarking by Template ranking with symmetrical Central pixel). AWTC insertion algorithm is:

1. Divide the binary image Z to be watermarked in a sequence v of non-overlapping pieces of image and sort v as in DHTC.

2. Clear the first n central pixels of the sorted sequence v, where n is the size of the adopted AS.

3. Using a cryptographically secure hashing function, compute the integrity-index H of the now-cleared image Z. Encrypt the integrity-index H with the secret- or private-key, obtaining the AS S.

4. Insert n bits of S in the n first central pixels of the sorted sequence v, obtaining the watermarked image.

AWTC verification is straightforward:

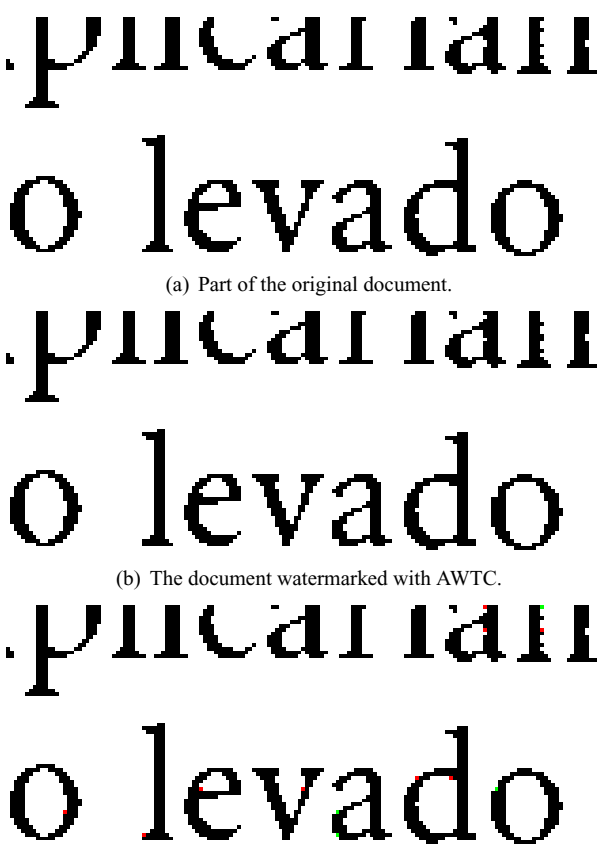
1. Divide the watermarked image Z in sequence v, and sort v as before.

2. Extract the AS S from the n first central pixels of the sorted v. Decrypt S with the secret- or public-key, obtaining the extracted integrity-index H.

3. Clear the first n central pixels of the sorted v and compute the check integrity-index C of the now-cleared image Z, using the hashing function.

4. If the extracted integrity-index H and the check integrity-index C are the same, the watermark is verified. Otherwise, the image was modified.

Figure 4(a) depicts a page of a magazine scanned at 300 dpi, resulting in a binary image with 3318×2536 pixels. This image was watermarked by AWTC with 10240-bits long MAC, resulting in figure 4(b). In order to test the visual quality of the watermarked image, we used a MAC 80 times



(a) Part of the original document.

(b) The document watermarked with AWTC.

(c) Flipped pixels are printed in color.

Fig. 4. A page of a magazine scanned at 300 dpi and watermarked by AWTC using an unusually long MAC.

longer than the usual MAC (128 bits) and 10 times longer than the usual RSA digital signature (1024 bits). Figure 4(c) depicts in red the white pixels that changed to black, and in green the black pixels that changed to white. It is easy to see that only the low visibility pixels have changed their values.

In AWTC (as in AWST), the number of data-bearing Z_1 pixels is exactly equal to the length of the adopted AS. All image pixels (except the n pixels that will bear the n bits of the AS) are taken into account to compute the AS. Consequently, parity attacks do not apply to AWTC. Moreover, *any* image alteration (even one single pixel flipping) can be detected.

5. CONCLUSIONS

In this paper, we have presented a new authentication watermarking for binary images named AWTC. AWTC can detect *any* alteration of the watermarked image with the security assured by the cryptography theory. Both secret- and public-key versions of AWTC are completely secure against parity attacks. Moreover, only the low-visibility pixels are flipped by the AWTC watermark insertion, resulting in watermarked images with excellent visual quality.

6. REFERENCES

[1] M. M. Yeung and F. Mintzer, "An Invisible Watermarking Technique for Image Verification," *IEEE Int. Conf. Image Processing*, 1997, vol. 1, pp. 680-683.

[2] P. W. Wong, "A Public Key Watermark for Image Verification and Authentication," *IEEE Int. Conf. Image Processing*, 1998, vol. 1, pp. 455-459, (MA11.07).

[3] P. W. Wong and N. Memon, "Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification," *IEEE Trans. Image Processing*, vol. 10, no. 10, pp. 1593-1601, 2001.

[4] P. S. L. M. Barreto, H. Y. Kim and V. Rijmen, "Toward a Secure Public-Key Blockwise Fragile Authentication Watermarking," *IEE Proc. Vision, Image and Signal Processing*, vol. 149, no. 2, pp. 57-62, 2002.

[5] M. U. Celik, G. Sharma, E. Saber and A. M. Tekalp, "Hierarchical Watermarking for Secure Image Authentication with Localization," *IEEE Trans. Image Processing*, vol. 11, no. 6, pp. 585-595, 2002.

[6] M. Wu, and B. Liu, "Data Hiding in Binary Image for Authentication and Annotation," *IEEE Trans. on Multimedia*, vol. 6, no. 4, pp. 528-538, 2004.

[7] M. S. Fu and O. C. Au, "Data Hiding Watermarking for Halftone Images," *IEEE Trans. Image Processing*, vol. 11, no. 4, pp. 477-484, 2002.

[8] Y.-C. Tseng, Y.-Y. Chen and H.-K. Pan, "A Secure Data Hiding Scheme for Binary Images," *IEEE Trans. on Communications*, vol. 50, no. 8, Aug. 2002, pp. 1227-1231.

[9] H. Y. Kim and A. Afif, "Secure Authentication Watermarking for Halftone and Binary Images," *Int. J. Imaging Systems and Technology*, vol. 14, no. 4, pp. 147-152, 2004.

[10] H. Y. Kim and R. L. Queiroz, "A Public-Key Authentication Watermarking for Binary Images," *IEEE Int. Conf. on Image Processing*, Singapore, 2004.

[11] H. Y. Kim and R. L. de Queiroz, "Alteration-Locating Authentication Watermarking for Binary Images," *Int. Workshop on Digital Watermarking 2004*, (Seoul), Lecture Notes in Computer Science 3304, pp. 125-136, 2004.

[12] R. de Queiroz and P. Fleckenstein, "Object Modification for Data Embedding through Template Ranking," *Xerox Invention Proposal*, 1999.