# A PUBLIC-KEY AUTHENTICATION WATERMARKING FOR BINARY IMAGES

*Hae Yong Kim*  and  Ricardo L. de Queiroz***
*Escola Politécnica, Universidade de São Paulo, Brazil. E-mail: hae@lps.usp.br.
**Universidade de Brasília, Brazil. E-mail: queiroz@ieee.org.

## ABSTRACT

Authentication watermarking is the process that inserts hidden data into an object (image) in order to detect any fraudulent alteration perpetrated by a malicious hacker. In the literature, quite a small number of secure authentication methods are available for binary images. This paper proposes a new secure authentication watermarking method for binary images. It can detect any visually significant alteration while maintaining good visual quality. As usual, the security of the algorithm lies on the secrecy of a private-key. Only its owner can insert the correct watermark while anyone may verify the authenticity through the corresponding public-key. A possible application of the proposed technique is in internet fax transmission, i.e. for legal authentication of documents routed outside the phone network.

## 1. INTRODUCTION

A watermark is a hidden signal added to images that can be detected or extracted later to make some assertion about the host image. Watermarking techniques can be classified as either "robust" or "fragile." Robust watermarks cannot be easily removed and should resist common image-manipulation procedures. They are useful for copyright and ownership assertion purposes. Fragile watermarks (or authentication watermarks) are easily corrupted by any image processing procedure. Watermarks for checking image integrity can be fragile because if the watermark is removed, the watermark detection algorithm will correctly report a corruption of the image.

In the literature, there are many authentication watermarking techniques (AWTs) for continuous-tone images [19, 18, 15, 16, 12, 2, 10, 3]. However, quite a small number of secure AWT is available for binary images. We mean by "secure AWT" a scheme where the security does not lie on the secrecy of the algorithm but only on the secrecy of the key. The watermarking algorithm and the fact that an image is watermarked may be made public without compromising the security. Hence, a secure AWT may rely upon cryptography. Moreover, a secure AWT must detect *any* visually significant change made to an image.

In a typical cryptography-based AWT an authentication signature (AS) is computed from the whole image data and inserted into the image itself. In cryptography, AS is called message authentication code (using secret-key) or digital signature (using public/private-key). AS contains information about the image contents that may be checked to verify its integrity. However, inserting the AS into the image alters the image itself, hence modifying its AS and invalidating the watermark. Typically, the information has to be somehow divided into at least two parts: a portion to maintain the image integrity and another portion to carry the AS. Although there are many papers on data hiding in binary images [13, 6, 1, 5, 4, 17, 7, 8, 14], there are only a few cryptography-based AWTs for binary images. Fu and Au [9] present a watermarking to detect unintentional changes in halftone images, but this cannot be considered to be an AWT because it does not withstand intentional or malicious attacks. Kim and Afif [11] present a cryptography-based AWT but it is especially suited for dispersed-dot halftone images and the visual quality for a generic binary image is poor. In this paper, we propose a new cryptography-based secure AWT for binary images that has good visual quality when applied to a generic binary image. It can be used in conjunction with secret-key or public/private-key cryptography. A possible use of our method is to send faxes and documents over uncontrolled networks and the internet. In this case, the receiver of a document can verify the document integrity (for a given originator).

## 2. DATA HIDING INTO BINARY IMAGES

As we noted in previous section, there are many data hiding techniques for binary images. Among them, an interesting technique, we name template ranking, was discovered by De Queiroz et al. [5] and independently by Wu et al. [17]. It can be applied to most binary images with excellent visual quality. It can be summarized as follows:

**1)** Divide the image $Z$ to be marked into small blocks (say, 8×8).
**2)** The neighborhood of each pixel (usually a 3×3 template) is analyzed to rate its visual significance. This is done via template ranking where the 512 3×3 binary templates are

ranked (uniquely associated with a number from 0 to 511) as in figure 1. Mirrors, transposes and reverses of each pattern have similar ranks, even though they were not shown. Also, there are a number of patterns that were not included for a particular reason: they might break existing objects and that might be objectionable in some applications. The reader can re-include them if necessary.

**3)** Insert one bit in each block by forcing the block to have even or odd number of white pixels, to insert bits 0 or 1 respectively. If the block already has the desired parity, it is left untouched. Otherwise, flip the pixel with the highest rank. This pixel is supposed to have the lowest visibility.

**4)** As different blocks may have different quantities of low visibility pixels, it is suggested to shuffle image $Z$ before embedding data.

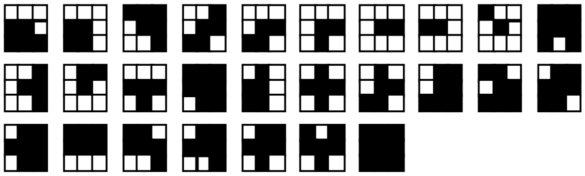

**Fig. 1:** 3×3 templates ranked from left to right, top to bottom. Mirrors, transposes and reverses of each pattern have the same rank, even though they were not included here. Templates not shown have lower ranking.

## 3. AUTHENTICATING BINARY IMAGES

The above data hiding technique can be employed in a secure AWT for binary images. That was hinted [17] but not elaborated before.

The idea of computing an AS of the whole binary image and inserting it into the same image fails because the insertion will modify the image fingerprint. The first idea to insert the AS without modifying the image fingerprint is to divide the image into two regions: the first (small) region where AS will be inserted, and the second (large) region from where AS will be computed. Let us write this idea algorithmically:

**1)** Let be given a binary image $Z$. Using a pseudo-random number generator with a seed, construct an auxiliary data structure called shuffling vector $V$, so that the image $Z$ can be viewed as a completely shuffled sequence of pixels $\tilde{Z}$. In the secret-key version of our technique, the secret-key is used as the seed of the pseudo-random generator. In the public/private-key version, the seed must be made public.

**2)** Let $n$ be the length of the adopted AS, and $m$ be the number of pixels of each block. Divide the shuffled sequence $\tilde{Z}$ into two regions:

- First region $\tilde{Z}_1$: $n \times m$ pixels, where AS is to be stored. This region is subdivided into $n$ blocks with $m$ pixels. In each block, one bit of AS will be inserted.
- Second region $\tilde{Z}_2$: the remainder of the shuffled sequence $\tilde{Z}$. The insertion algorithm will compute AS of this region.

**3)** Using a cryptographically secure hashing function $H$, compute the fingerprint of the second region $H = H(\tilde{Z}_2)$. Cryptograph the fingerprint $H$ using a secret- or private-key, obtaining an AS: $S = K(H)$.

**4)** Insert $S$ into the first region, obtaining the watermarked image $Z'$. Insert one bit of $S$ in each block as previously described.

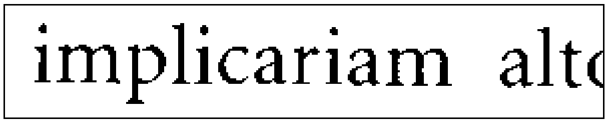

**Fig. 2a:** Part of a page of magazine scanned at 300 dpi.

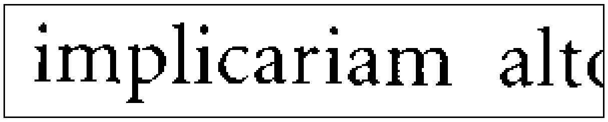

**Fig. 2b:** Part of image with 1024 bits embedded (enough for RSA digital signature).

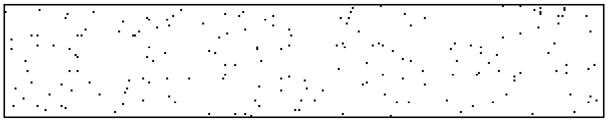

**Fig. 2c:** Region defining map: black pixels belong to region 1, while white ones to region 2.

We remark here that AS cannot be made too short without compromising seriously the security. Usually, a message authentication code 128 bits long is considered secure. The most well-known digital signature, RSA, is considered secure with 1024 bits. A newer scheme, DSA, is considered secure with 320 bits. The verification algorithm of a watermarked image $Z'$ is straightforward:

**1)** Compute the same shuffling vector $V$ used in the insertion. Note that in the secret-key version, the secret-key is also the seed of the random number generator and consequently only the owner of the key can reconstruct the shuffling vector. However, in the public-key version, the seed is public and consequently the shuffling vector is public.

**2)** Divide $Z'$ in two regions $Z'_1$ and $Z'_2$, in the same way as done in the insertion step. Compute the fingerprint $H$ of $Z'_2$.

**3)** Extract the AS stored in $Z'_1$ and decrypt it using the secret- or public-key, obtaining the check data $D$.

**4)** If $D = H$, the watermark is verified. Otherwise, image $Z'$ was modified or a wrong key was used.

Figure 2(a) depicts a zoom of a magazine page scanned at 300 dpi, which can be considered a “typical” binary document. Figure 2(b) is the corresponding image after embedding 1024 bits using 64-pixel blocks.

## 4. PARITY ATTACK

The proposed method detects *any* alteration perpetrated to the second region of a watermarked image, even to a single pixel. Indeed, the probability of not detecting an alteration in this region is only $2^{-n}$, which can be neglected (*n* is the length of the AS).

Unfortunately, any alteration that maintains the parities of blocks in the first region cannot be detected. For example, if two pixels that belong to the same block change their values, the parity of this block does not change and this modification will pass undetected. We named this a "parity attack." If the watermarked image $Z'$ is large enough, the pixels of $Z'_1$ constitute isolated pixels randomly dispersed in the image $Z'$ and it is unlikely that a malicious attacker can introduce any visually significant alteration in $Z'$, changing only pixels of $Z'_1$ (while maintaining the parity of all blocks). For example, in figure 2c, black pixels belong to region 1 and they are quite dispersed. Thus no visually meaningful alteration will result by modifying only region 1 pixels.

However, if the image $Z'$ is small, the pixels of $Z'_1$ can form contiguous regions in $Z'$, what arises the possibility of successful visually significant modifications that will pass undetected. For example, figure 3(a) is an image of a ballot ticket and figure 3(b) is the same image after marking. Figure 3(c) shows pixels that belong to region 1 in black. Any region 1 pixel can be modified, provided that another pixel in the same block is also modified. To obtain a faked image, a malicious hacker changes a pixel *p* of block *i*. Then, he looks for the pixel within the block *i* with the highest rank and flips its value. Figure 3(d) shows a faked image constructed by repeating this idea. This alteration will pass undetected by the proposed algorithm.

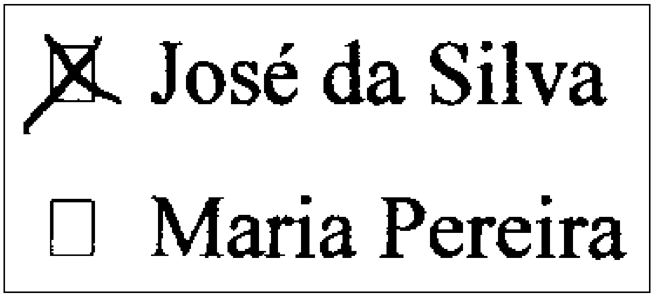

**Fig. 3a:** A small image (370×160) to be watermarked.

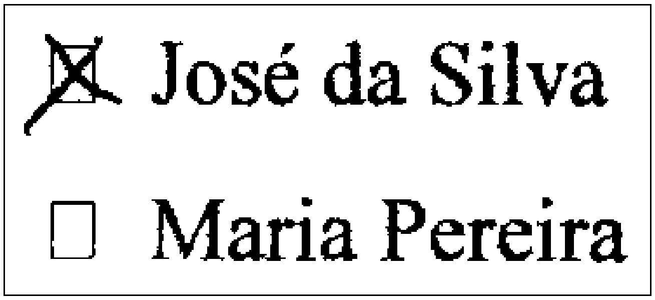

**Fig. 3b:** Image with 800 bits embedded using blocks of 64 pixels.

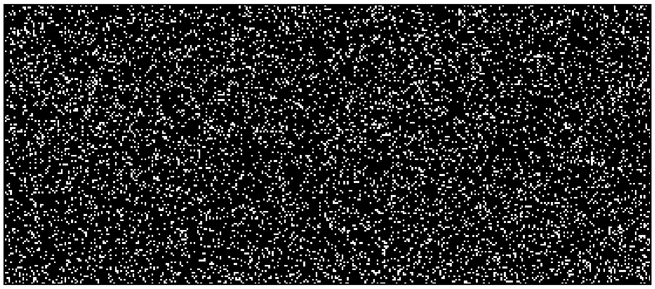

**Fig. 3c:** Black pixels belong to region 1, where parity attack can be mounted.

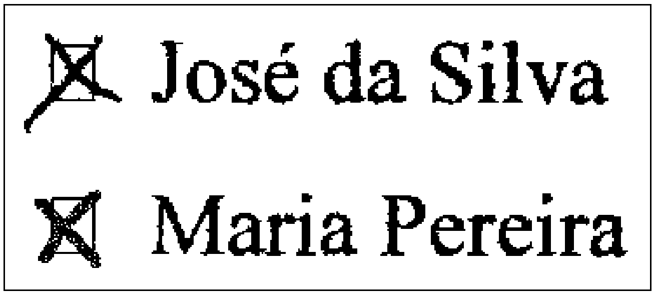

**Fig. 3d:** Faked image generated by the parity attack, undetectable by the first version of our algorithm.

Actually, the scenario described above only applies to the public-key version of the algorithm, where the locations of regions 1 and 2, as well as the subdivisions of region 1 into blocks are publicly known. In the secret-key version of the algorithm, we do not have to worry *much* about a parity attack, because the secret-key is used to generate the shuffling vector. So, an attacker will not know how the watermarked image is divided into regions 1 and 2, and how region 1 is subdivided into blocks. However, we have to pay some attention to this potential attack because the attacker may have many different ways to obtain "clues" about the location of regions and blocks. For example, let us suppose that the attacker has access to a database of original and marked binary documents, all of the same size and all watermarked using the same secret-key. Then it will be known that all those pixels whose values are different in the original and the watermarked version images belong to region 1.

## 5. IMPROVED ALGORITHM TO DEFEND AGAINST PARITY ATTACKS

In order to minimize the possibility of a parity attack, we suggest the following improvement of the step 4 of the insertion algorithm:

**4)** Insert *S* into the first region using the following algorithm, obtaining the watermarked image $Z'$:

```
For $i = 0$ to $n$-1 {
    Insert bit $i$ of $S$ into block $i$ forcing it to have odd/even number or white pixels;
    Compute new AS $S$, feeding hashing function with the content of block $i$ and cryptographing it:
        $S \leftarrow K(H(S, \text{pixels of block } i))$ ;
}
```

This way, block *n*-1 still can suffer a parity attack. However, if block *n*-2 is modified without modifying its parity, with 50% of chance this modification will be detected. If block *n*-3 is modified (maintaining its parity), there is 75% of chance of detecting this change. If block 0 is changed (maintaining its parity), there is a probability of $1-2^{-(n-1)}$ of detecting this change. The improved algorithm certainly makes it much more difficult to be subject to a parity attack.

## 6. CONCLUSIONS

This paper has proposed a new cryptographically secure authentication watermarking for binary images. It can be used with secret- or public/private-key cipher. We also derived a variant of the algorithm that is able to withstand parity attacks. The proposed technique is suitable to watermark most binary images except dispersed-dot halftones. The method using a public/private-key can be used in trusted FAX machines, i.e. to electronically sign binary documents. Further research is necessary to adapt the method to scanned documents.

## 7. ACKNOWLEDGEMENT

One of authors, H. Y. Kim, would like to express his gratitude to CNPq for the partial financial support of this work under grant 300689/98-5.

## 8. REFERENCES

[1] Z. Baharav and D. Shaked, "Watermarking of Dither Halftone Images", Hewlett-Packard Labs. Tech. Rep. HPL-98-32 (1998).

[2] P. S. L. M. Barreto and H. Y. Kim, "Pitfalls in Public Key Watermarking," *Sibgrapi - Brazilian Symp. Computer Graphics and Image Processing*, 1999, pp. 241-242.

[3] 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.

[4] Y.-Y. Chen, H.-K. Pan and Y.-C. Tseng, "A Secure Data Hiding Scheme for Binary Images," *IEEE Symposium on Computers and Communications*, 2000, pp. 750-755.

[5] R. de Queiroz and P. Fleckenstein, "Object Modification for Data Embedding through Template Ranking," Xerox Internal Document, 1999.

[6] M. P. Deseilligny and H. Le Men, "An Algorithm for Digital Watermarking of Binary Images, Application to Map and Text Images," available at www-ima.enst.fr/~maitre/tatouage/MPdS_HK.ps, 1998.

[7] M. S. Fu and O. C. Au, "Data Hiding by Smart Pair Toggling for Halftone Images," *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp. 2318-2321, 2000.

[8] 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.

[9] M. S. Fu and O. C. Au, "A Robust Public Watermark for Halftone Images," *IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 639-642, 2002.

[10] M. Holliman and N. Memon "Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes," *IEEE Trans. Image Processing*, 2000, vol. 9. no. 3, pp. 432-441.

[11] H. Y. Kim and A. Afif, "Secure Authentication Watermarking for Binary Images," in *Proc. Sibgrapi - Brazilian Symp. on Comp. Graph. and Image Proc.*, pp. 199-206, 2003.

[12] C. T. Li, D. C. Lou and T. H. Chen, "Image Authentication and Integrity Verification via Content-Based Watermarks and a Public Key Cryptosystem," *IEEE Int. Conf. Image Processing*, 2000, vol. 3, pp. 694-697.

[13] N. F. Maxemchuk and S. Low, "Marking Text Documents," *Int. Conf. Image Processing*, vol. 3, pp. 13-17, 1997.

[14] 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. 1227-31.

[15] P. W. Wong, "A Watermark for Image Integrity and Ownership Verification," *IS&T PIC Conference*, (Portland, OR), May 1998 (also available as Hewlett-Packard Labs. Tech. Rep. HPL-97-72, May 1997).

[16] 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).

[17] M. Wu, E. Tang and B. Liu, "Data Hiding in Digital Binary Image," *IEEE Int. Conf. Multimedia and Expo*, ICME'00, New York, USA, 2000.

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

[19] J. Zhao and E. Koch, "Embedding Robust Labels into Images for Copyright Protection," *Proc. Int. Cong. Intellectual Property Rights, Knowledge and New Technologies*, 1995, pp. 242-251.