A Secure Authentication Watermarking for Halftone and Binary Images

Hae Yong Kim, Amir Afif

Universidade de São Paulo, Escola Politécnica,

Av. Prof. Luciano Gualberto, trav. 3, 158, CEP 05508-900, São Paulo, SP, Brazil.

hae@lps.usp.br, amir@usp.br

Abstract. Authentication watermark is a hidden data inserted into an image that can be used to detect any accidental or malicious alteration in the image. In the literature, there are many authentication-watermarking techniques for continuous-tone images. However, quite a small number of secure authentication watermarking techniques are available for binary/halftone images. This paper proposes a simple solution for inserting a secure authentication watermark in binary/halftone images. It consists on choosing a set of pseudo-random pixels in the image, clearing them, computing the message authentication code (or the digital signature) of the random-pixels-cleared image, and inserting the resulting code into the selected random pixels. Dispersed-dot halftone images watermarked by the proposed technique present better visual quality than watermarked generic binary images. However, in practice, the size of the host image uses to be much larger than the size of the inserted code, and consequently the visual degradation is hardly noticeable in either case. The proposed technique seems to be the only binary/halftone watermarking scheme that can detect even a single pixel alteration in the host image. It can be used with secret-key or public-key ciphers.

1. Introduction

A watermark is a signal added to digital data (audio, video and still images) that can be extracted later to make an assertion about the data. In this paper, we will be concerned only with still images. Digital watermarking techniques can be classified as either “robust” or “fragile.”

Robust watermarks are designed to be hard to remove and to 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 the image integrity can be fragile because if the watermark is removed, the watermark detection algorithm will correctly report the corruption of the image.

Halftoning is a process used to convert a grayscale image G into the corresponding binary image B, such that B looks like G when viewed from a distance. Classic halftone methods include ordered dithering, error diffusion (Ulichney, 1987) and dot diffusion (Knuth, 1987). Halftone images appear routinely in books, magazines, newspapers, printer outputs, and fax documents. Halftone images can be dispersed-dot or clustered-dot. Usually, dispersed-dot halftone images present a better visual quality, but some devices cannot reproduce too-finely-dispersed dots (like laser printers) and they must use clustered-dot halftoning.

In the literature, there are many authentication-watermarking techniques for continuous-tone images (Zhao and Koch, 1995; Yeung and Mintzer, 1997; Wong, 1997; Wong 1998; Li et al., 2000; Barreto and Kim, 1999; Holliman and Memon, 2000; Barreto et al., 2002). However, it seems to be very difficult to design a really secure authentication watermarking without making use of the solid cryptography theory and techniques. Indeed, those watermarking techniques that were not founded in cryptography theory (Zhao and Koch, 1995; Yeung and Mintzer, 1997) or those that applied cryptographic techniques without the due care (Wong, 1997; Wong, 1998; Li et al., 2000) were later shown to be unreliable (Holliman amd Memon, 2000; Barreto et al., 2002).

In a cryptography-based authentication watermarking, the message authentication code (MAC) or the digital signature (DS) of the whole image is computed and inserted into the image itself. However, inserting the MAC/DS alters the image and consequently alters its MAC/DS, invalidat-

ing the watermark. To avoid this problem, for gray-scale and color images, usually the least significant bits (LSBs) are cleared, the MAC/DS of the LSB-cleared image is computed and then the code is inserted into the LSBs. In other words, those bits where the watermark is to be inserted are not taken into account when computing the MAC/DS.

For binary and halftone images, this idea fails completely, because each pixel has only one bit. Modifying any pixel in order to embed watermarking modifies the fingerprint of the image and invalidates the watermarking. Consequently, although there are many papers on data hiding in binary images (Maxemchuk and Low, 1997; Baharav and Shaked, 1998; Chen et al., 2000; Wu et al., 2000; Fu and Au, 2000; Hel-Or, 2001, Fu and Au, 2002a; Pei and Guo, 2003; Chun and Ha, 2003), we have no tidings of a cryptography-based authentication watermarking for binary and halftone images. Fu and Au (2002b) present a watermarking to detect unintentional changes in halftone images, but this cannot be considered to be an authentication watermarking because it does not withstand intentional or malicious attacks.

After the well-accepted cryptography paradigm, the security of an authentication watermarking must lie only on the secrecy of the key. The fact that the image is watermarked as well as the watermarking algorithm may be public without compromising the security of the scheme. In this paper, we propose an authentication watermarking for halftone and binary images that complies with this requirement. The idea is quite simple. However, it seems to be a new idea (it was published in our earlier conference paper (Kim and Afif, 2003)). The main idea consists on clearing all those pixels of the host image that can be modified by the insertion of the watermark, before computing the MAC/DS. More precisely, it consists on choosing a set of pseudo-random pixels in the image, clearing them, computing the MAC/DS of the random-pixels-cleared image, and inserting the resulting code into the selected random pixels.

Dispersed-dot halftone images watermarked by this technique present better visual quality than watermarked generic binary images (including clustered-dot halftone images). However, in practice, the size of the host image uses to be much larger than the size of the inserted code, and consequently the visual degradation is hardly noticeable in either case. The size of MAC/DS is independent of the size of the host image. Consequently, the smaller the host image, the more visually noticeable will be the watermark insertion.

The proposed technique can be used in conjunction with a symmetric cipher (secret-key cryp-

tography) or an asymmetric cipher (public-key cryptography) to detect *any* alteration of the watermarked image, even one single pixel toggling. The technique is intended to authenticate only digital binary images. Printed images cannot be authenticated using the proposed technique.

Friedman (1993) introduced the concept of "trustworthy digital camera." This concept can be straightforwardly adapted to fax machines, using the proposed technique. Using a "trustworthy fax," the receiver of a document can certify the originator of the document, and that it has not been altered (accidentally or maliciously) during the transmission.

2. Data Hiding in Binary and Halftone Images

There are many papers in the literature for inserting a sequence of bits in binary and halftone images. They can be divided in three basic classes:

1 - Pixel-wise: Change the values of (usually pseudo-randomly chosen) individual pixels (Fu and Au, 2000; Fu and Au, 2002a; Chun and Ha, 2003).

2 - Component-wise: Change the characteristics of pixel groups, for example the thickness of strokes, the position or the area of connected components, etc. (Maxemchuk and Low, 1997). Unfortunately, the success of this approach depends highly on the type of the host image.

3 - Block-wise: Divide the host image into blocks and modify some characteristic of each block. Some works (Baharav and Shaked, 1998; Hel-Or, 2001; Pei and Guo, 2003) suggest alternating between two different dithering matrices to halftone a image such that the matrix used to halftone each block can be determined in the future by analyzing the statistical properties. Other works suggest modifying slightly the content of the block so that it hides the desired sequence of bits (Wu et al., 2000; Tseng et al., 2002).

DHST (data hiding by self toggling) is a pixel-wise data hiding technique, and it is especially interesting for its simplicity (Fu and Au, 2000; Fu and Au, 2002a). It was originally designed to embed bits in dispersed-dot halftone images. In DHST, a pseudo-random number generator with a known seed is used to generate a set of non-repeating pseudo-random locations within the image. Then one bit is embedded in each location by forcing it to be either black or white. With probability 50%, the pixel of the original halftone is the desired value and thus no change is needed. And with probability 50% the pixel is opposite to that of the desired value, and the pixel needs to be

altered. To read the hidden data, one simply uses the same random number generator and the same seed to obtain again the non-repeating pseudo-random locations. Then the pixel values at those locations can be read easily. Evidently, DHST can also be used to embed data in any binary image. However, in this case, a visible salt-and-pepper noise will appear. In this paper, we will transform DHST into a cryptographically secure fragile authentication watermarking. Fu and Au (2000; 2002a) present many improvements to this basic idea to enhance the visual quality of the host image: DHPT (data hiding by pair toggling) and DHSPT (data hiding by smart pair toggling). The underlying idea of these improvements is to keep constant the local average intensity. At the selected pseudo-random locations, the pixel alteration is accompanied by a complementary modification of a neighbor.

3. Authentication Watermark for Continuous-Tone Images

In the literature, most works on authentication watermarking are designed for continuous-tone images. Their goal is not only to detect alteration in the host image but also to spatially locate them. Papers (Holliman and Memon, 2000; Barreto et al., 2002) analyze thoroughly the requirements to obtain a reliable alteration-locating authentication-watermarking technique. However, in the present paper, we will simply detect the alteration, without locating it. The reason for this simplification is that binary images present a data hiding capacity much smaller than continuous-tone images.

A cryptography-based authentication watermarking (Wong, 1998; Holliman and Memon, 2000; Barreto et al., 2002) typically performs the following operations for the watermark insertion:

1. Let Z be a grayscale image to be watermarked and let A be a logo binary image to be inserted into Z.

2. Let Z^* be the image obtained from Z by clearing the LSBs of all pixels. Using a cryptographically secure hashing function H, compute the image fingerprint $H = H(Z^*)$.

3. Exclusive-or H with A, getting the marked fingerprint $\hat{H}$.

4. Encrypt $\hat{H}$ with the secret-key (symmetric cipher) or private-key (asymmetric cipher), thus generating a MAC/DS S.

5. Insert *S* into the LSBs of *Z**, obtaining the marked image Z'.

The watermark-verifying algorithm is:

1. Let *X*' be a watermarked image. Let *X** be the image obtained from *X*' by clearing the LSBs of all pixels. Using the same hashing function *H* chosen for the insertion, compute the image fingerprint $H = H(X^*)$.

2. Extract the LSBs from *X*' and decrypt the result using the secret-key (symmetric cipher) or public-key (asymmetric cipher), obtaining the decrypted data *D*.

3. Exclusive-or *H* with *D*, obtaining the check image *C*.

4. If *C* and *A* are equal, the watermark is verified. Otherwise, the marked image *X*' has been modified.

Notice that, theoretically, the image *A* must be publicly available for the verification to take place. In practice, however, *A* is a meaningful logo image and any change in *X*' will most likely generate a noise-like image *C*, which cannot be mixed up with *A*, even if *A* is not publicly available. Moreover, *A* can be a very simple image, like an entirely white or black image, without compromising the security of the scheme.

4. The Proposed Method

As we said in section 1, for binary/halftone images, the insertion of the watermark changes the host binary image *B* and consequently changes its fingerprint. That is, $H(B) \neq H(B')$. How can we overcome this problem?

We present a very simple solution using the DHST scheme. Differently from most binary image data hiding techniques, in DHST only a few pixels are modified and the precise positions of those pixels are known both in the insertion and extraction phases. Consequently, these pixels can be cleared before computing the hashing function, just like clearing LSBs for grayscale images. Let us call the obtained technique AWST (authentication watermarking by self toggling). The AWST insertion algorithm is:

1. Let *B* be a binary image to be watermarked and let *A* be a logo binary image to be inserted into

B.

2. Use a pseudo-random number generator with a known seed to generate a set of non-repeating pseudo-random locations *L* within the image *B*.

3. Clear all pixels of *B* that belong to *L*, obtaining *B**.

4. Compute the fingerprint $H = H(B^*)$.

5. Exclusive-or *H* with *A*, getting the marked fingerprint $\hat{H}$.

6. Encrypt $\hat{H}$ with the secret-key (symmetric cipher) or private-key (asymmetric cipher), thus generating the MAC/DS *S*.

7. Insert *S* into the set of pixels *L*, generating the watermarked image *B*'.

The AWST extraction algorithm is:

1. Let *X*' be a watermarked image. Use the same pseudo-random number generator and the same seed to generate again the same set of non-repeating pseudo-random locations *L* where the watermark has been inserted.

2. Let *X** be the image obtained from *X*' by clearing all pixels in *L*. Using the same hashing function *H* chosen for the insertion, compute the fingerprint $H = H(X^*)$.

3. Extract the watermark from *X*' by scanning pixels in *L* and decrypt the result using the secret-key (symmetric cipher) or public-key (asymmetric cipher), obtaining the decrypted data *D*.

4. Exclusive-or *H* with *D*, obtaining the check image *C*.

5. If *C* and *A* are equal, the watermark is verified. Otherwise, the marked image *X*' has been modified (or a wrong key has been used).

Figure 1 illustrates the AWST watermarking scheme. Let us suppose that image *B* (figure 1a) is a sensitive image to be transmitted through an unreliable channel, where unintentional or intentional alterations may occur. To protect *B*, a logo image *A* (figure 1b) was inserted into *B* using the AWST algorithm. Image *B*' (figure 1c) is the watermarked image where 1024 bits were embedded. This is enough to embed a RSA digital signature. If the extraction algorithm is performed, we ob-

tain the check image *C* (figure 1d), exactly equal to the logo image *A*. If even a single pixel of *B*' is altered, the extracted image is completely noisy (figure 1f).

Figure 2 depicts the quality of an AWST-watermarked document. A page of a magazine was scanned at 300 dpi, resulting a "typical" binary document with 3318 rows and 2536 columns (figure 2a). Figures 2b, 2c and 2d show respectively the document with 128 bits, 320 bits and 1024 bits embedded. These quantities of bits are enough to insert, respectively, a secret-key message authentication code, a public-key DSA digital signature and a public-key RSA digital signature (Schneier, 1996). The visual qualities of the watermarked documents are excellent, because the quantities of inserted bits are insignificant compared to the number of pixels of the host image.

5. Variations on the Proposed Method

Fragile authentication watermarks can be subdivided into three subcategories: keyless, secret-key and public-key watermarks. All three subcategories can be obtained using the AWST and they are described in subsections 5.1, 5.2 and 5.3.

Another possible variation is a watermarking scheme that does not use the logo image. In this case, the detection algorithm will not extract a check image *C*, but it will answer a Boolean question: the image has or has not been altered (subsection 5.4).

Another possible variation is to use the improved data hiding techniques DHPT and DHSPT (instead of the DHST) to insert the watermark in dispersed-dot halftone images, keeping constant the average grayscale around the pseudo-random pixels (subsection 5.5).

5.1 Keyless AWST

Keyless authentication watermarking is useful for detecting unintentional alterations in images such as cropping and distortions due to dirt or human writing/marking. It is a sort of "check-sum". If the watermarking insertion and detection algorithms are made public, anyone can insert and verify keyless authentication watermarks. In the keyless AWST, the seed of the pseudo-random number generator must be made public. The encryption (step 6 of the AWST insertion algorithm) must be eliminated as well as the decryption (step 3 of the AWST extraction algorithm). The hashing function can be very short, say 24 bits, because it is very unlikely that an unintentional alteration

will cause a hashing collision.

5.2 Secret-Key AWST

Secret-key authentication watermarking can be used to detect any alteration in image, even intentional or malicious ones. This kind of watermarking is similar to the well-known message authentication code. The only difference is that the authentication code is inserted into the host image instead of being independently stored. Algorithms for watermark insertion and detection can be public and a secret-key is used in both phases. The seed of the pseudo-random number generator may be kept secret or made public, because the security does not lie on the secrecy of the seed.

Let us suppose that Alice administers a large image database where each image is signed with a secret-key *k* that only Alice knows. Let us suppose that Mallory, a malicious active attacker, modifies one image in the database. Mallory cannot insert the correct watermark into the modified image because he does not know *k*. Moreover, Alice will be able to detect all images that were altered by Mallory using the AWST extraction algorithm and her secret-key *k*.

Typically, a 128-bits long message authentication code (MAC) is considered cryptographically secure. Introductory books on cryptography, like (Schneier, 1996), explain many different MAC schemes.

5.3 Public-Key AWST

Public-key authentication watermarking techniques use a public-key cryptography to insert the digital signature into the host image. Using a public-key cipher, claims of image authenticity can be judged without the necessity of disclosing any private information.

Let us suppose that Alice wants to send to Bob a sensitive image without disclosing her secret-key. Alice uses her private-key to insert watermark into the image and sends it to Bob. Bob uses Alice's public-key to verify that Alice signed the image and nobody introduced any alteration after Alice signing it. If Mallory, a malicious hacker, alters the image, he cannot insert the correct watermark into the falsified image because he does not know Alice's private-key.

Public-key authentication watermarking for binary images can be used, for example, in secure fax transmission. Let us suppose that each fax machine has its own internal secret-key. Each time a

transmission is performed, the sending fax machine inserts the AWST watermark. The fax receiver knows the public-key of the sending machine. Thus, the receiver can verify that a specific fax machine originated the document and that the document has not been manipulated. The verification takes place using the digital image received by the fax machine. The printed version of the image cannot be authenticated using AWST.

Another possible application of the public-key AWST is in legal usage of binary documents. If documents are transmitted through the Internet, it is important that the receiver ensures that a specific person has signed the document and that no alteration has occurred after the signature.

A database of sensitive documents can be protected against fraudulent manipulations using the public-key AWST and anyone can verify the authenticity of a document using the public-key.

The most well known digital signature, RSA, is considered secure with 1024 bits. A newer scheme, DSA, is considered secure with 320 bits (Schneier, 1996).

5.4 Boolean Answer

Although extracting a visible logo from a watermarked image may be appealing, actually we only need to receive a binary answer from the AWST extraction algorithm: whether the image contains a valid watermark or not. To obtain a Boolean answer, we can eliminate step 5 from the AWST insertion algorithm and step 4 from the AWST extraction algorithm.

5.5 Keeping Constant Local Average Grayscales

The visual quality of an AWST-marked dispersed-dot halftone image can be improved by using improved data hiding techniques DHPT and DHSPT (Fu and Au, 2000; Fu and Au, 2002a), instead of DHST. These improvements try to keep constant the local average intensity. At the selected pseudo-random locations, the pixel alteration is accompanied by a complementary modification of a neighbor pixel.

However, to implement this scheme, no neighbor of the selected pseudo-random pixels must be fed into the hashing function. Consequently, these locations will remain unprotected, that is, if an alteration occurs in one of neighbors of a selected pseudo-random pixel, this alteration will not be detected by the AWST scheme.

6. Conclusions

This paper has proposed a cryptographically secure authentication-watermarking technique for halftone and binary images named AWST. It is especially suitable for authenticating dispersed-dot halftone images. However, as the quantity of pixels to be altered is very small (compared to the size of a "typical" binary document), it can be applied to any binary image without causing a considerable loss of quality. The proposed technique can be used in three ways: keyless, secret-key and public-key. Public-key AWST is the most useful. Public-key AWST can be used in trustworthy fax machines, to electronically sign binary documents, to protect a database of sensitive documents, etc.

7. Acknowledgements

The authors would like to express their gratitude to FAPESP and CNPq for the partial financial supports of this work under grants 2001/02400-9 and 300689/98-5, respectively.

8. References

Z. Baharav, and D. Shaked, Watermarking of Dither Halftone Images, Hewlett-Packard Labs Tech Rep, HPL-98-32, 1998.

P.S.L.M. Barreto, and H.Y. Kim, Pitfalls in Public Key Watermarking, Proc Brazilian Symp on Computer Graphics and Image Processing, 1999, pp. 241-242.

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 2 (2002), 57-62.

Y.Y. Chen, H.K. Pan, and Y.C. Tseng, A Secure Data Hiding Scheme for Binary Images, IEEE Symp Computers and Communications, 2000, pp. 750-755.

I.G. Chun and S. Ha, A Robust Printed Image Watermarking Based on Iterative Halftoning Method, Second Int Workshop on Digital Watermarking, Lecture Notes on Computer Science, no. 2939, 2003, pp. 200-211.

G.L. Friedman, The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image, IEEE Trans Consumer Electronics (1993), 905-910.

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

M.S. Fu, and O.C. Au, Data Hiding in Halftone Images by Stochastic Error Diffusion, IEEE Int Conf Acoustics Speech and Signal Processing, 2001, pp. 1965-1968.

M.S. Fu, and O.C. Au, Data Hiding Watermarking for Halftone Images, IEEE Trans Image Processing 4 (2002a), 477- 484.

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

H. Z. Hel-Or, Watermarking and Copyright Labeling of Printed Images, Journal of Electronic Imaging 3 (2001), 794-803.

M. Holliman, and N. Memon, Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes, IEEE Trans Image Processing 3 (2000) 432-441.

H.Y. Kim, and A. Afif, Secure Authentication Watermarking for Binary Images, Proc Brazilian Symp on Computer Graphics and Image Processing, 2003, pp. 199-206.

D.E. Knuth, Digital Halftones by Dot Diffusion, ACM Trans Graphics 4 (1987) 245-273.

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.

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

S.C. Pei, and J.M. Guo, Hybrid Pixel-Based Data Hiding and Block-Based Watermarking for Error-Diffused Halftone Images, IEEE Trans Circuits and Systems for Video Technology, 8 (2003) 867-884.

B. Schneier, Applied Cryptography, second edition, John Wiley & Sons, 1996.

Y.C. Tseng, Y.Y. Chen, and H.K. Pan, A Secure Data Hiding Scheme for Binary Images, IEEE Trans Communications, 8 (2002) 1227-1231.

R. Ulichney, Digital Halftoning, The MIT Press, 1987.

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

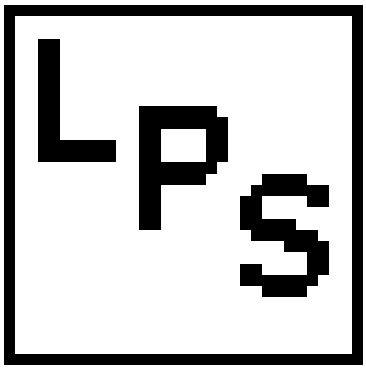
P.W. Wong, A Public Key Watermark for Image Verification and Authentication, IEEE Int Conf Image Processing, 1998, vol. 1, pp. 455-459.

M. Wu, E. Tang, and B. Liu, Data Hiding in Digital Binary Image, IEEE Int Conf Multimedia and Expo, New York, USA, 2000, 393-396.

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

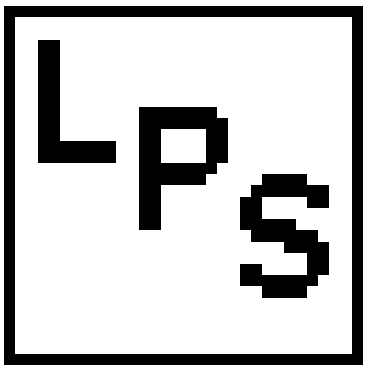
J. Zhao and E. Koch, Embedding Robust Labels into Images for Copyright Protection, Int Cong Intellectual Property Rights, Knowledge and New Technologies, 1995, 242-251.

(1a) Part of a 512×512 dispersed-dot halftone image *B* to-be-watermarked.

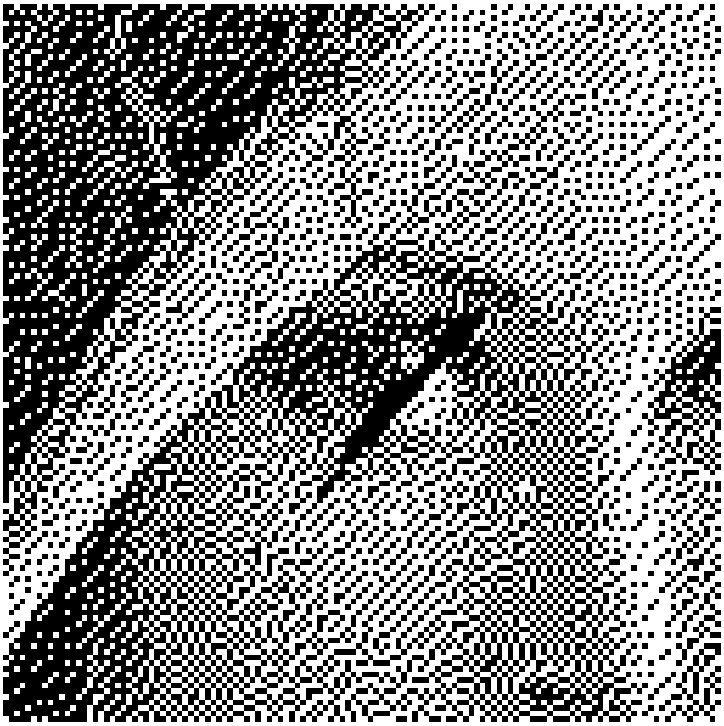


(1b) Logo image *A* (32×32 pixels) to-be-inserted into *B*.

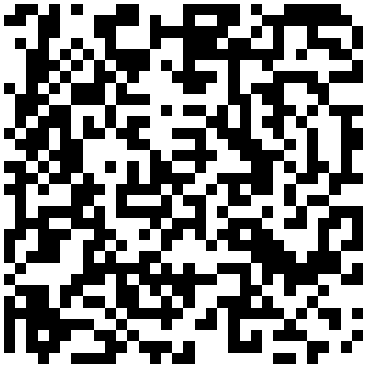
(1c) Part of watermarked image *B*'. 1024 bits were embedded.



(1d) Check image *C* extracted from image *B*'.

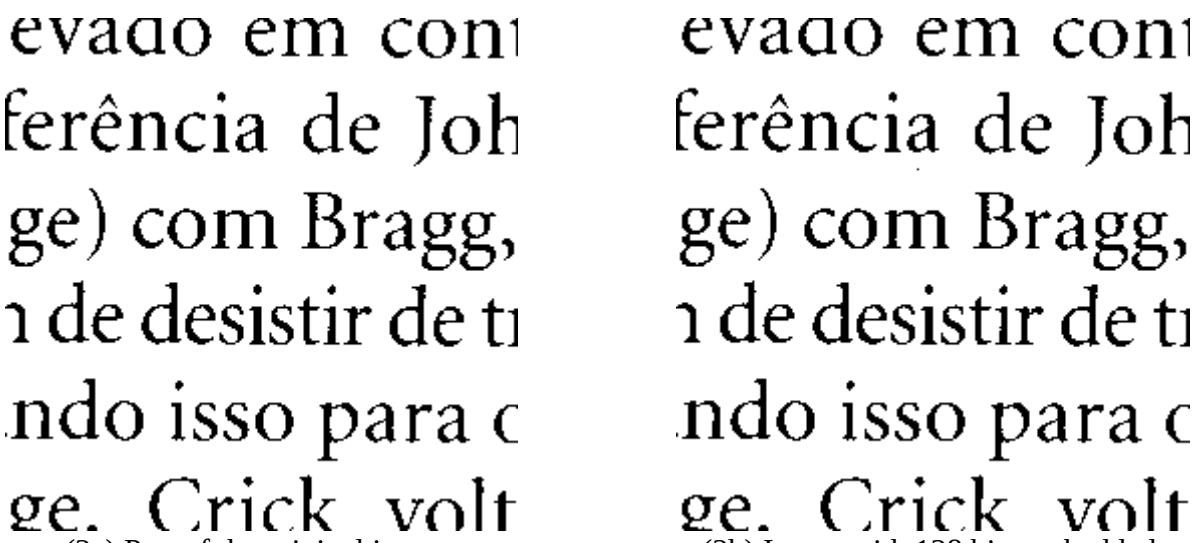


(1e) Part of the modified image *X'*.

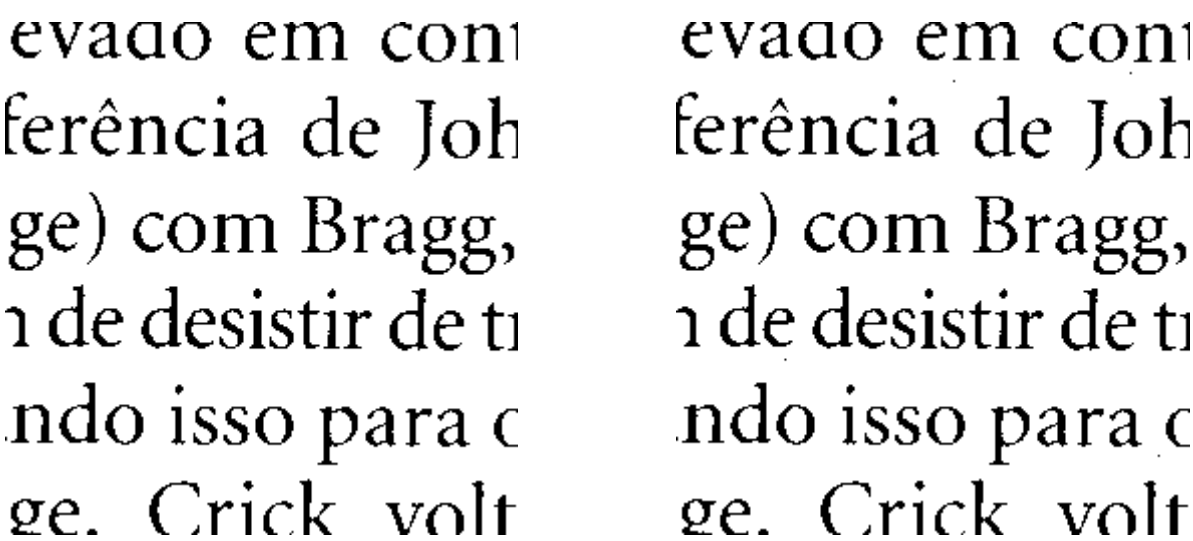


(1f) Check image *C* extracted from image *X'*.

Fig. 1: Illustration of the public-key AWST. Logo image *A* (1b) was inserted into image *B* (1a) using a public-key cipher. Figure (1c) depicts the watermarked image. Executing the watermark extraction algorithm, the correct check image *C* (1d) was obtained. When the watermarked image was modified (1e), a completely random check image was extracted (1f).



(2a) Part of the original image. (2b) Image with 128 bits embedded.



(2c) Image with 320 bits embedded. (2d) Image with 1024 bits embedded.

Fig. 2: The quality of a "typical" AWST-watermarked document. (2a) A page of a magazine was scanned at 300 dpi, resulting a binary image with 3318 rows and 2536 columns. (2b) Image marked with 128-bit secret-key message authentication code. (2c) Image marked with 320-bit public-key DSA signature. (2d) Image marked with 1024-bit public-key RSA signature.