

A Watermark for Image Integrity and Ownership Verification

Ping Wah Wong
Hewlett Packard Company, 11000 Wolfe Road, Cupertino, CA 95014

Abstract

We describe in this paper a watermarking scheme for ownership verification and image authentication. This authentication watermark is invisible, and has applications in trusted cameras, image transactions, legal usages, medical archiving, and others. This watermark, in conjunction with an appropriate digital key, can detect and localize any modifications to images. The watermark inherits the security of a cryptographic hash function, hence it is computationally infeasible for any unauthorized user to confuse the image integrity or ownership by forgery. We also describe using this invisible authentication watermark to protect any visible watermarks.

Introduction

Digital watermarking is a technique for inserting a digital signature (watermark) into an image, where the signature can be extracted or detected for identification or authentication purposes. There are different types of watermarks that are designed for different applications [1]. For example, ownership assertion watermarks can be inserted to images that are to be posted publicly, so that unauthorized users who claim ownership or resell the images without the consent of the original owner can be caught. This type of watermark is typically robust [2–5] in that the watermark should still be detectable after the watermarked image has been processed by common image processing algorithms such as image scaling, cropping, and compression.

It is well known that digital images can be altered or manipulated with ease. Furthermore, it is generally impossible to tell whether the altered image or the original image is the authentic one. This is an important issue in, for example, news reporting or in legal usages, where we want to be sure a digital image truely reflects what the scene looks like. Another need of image authentication arises in, for example, electronic commerce where a buyer buys a digital image from a seller, and then the seller transmits the digital image to the buyer over the network. In this case the buyer wants to ensure that the received image was indeed the genuine image sent by the seller. Here we not only want to verify the integrity of an image, we also want to check the original ownership.

Previously, the idea of a trusted digital camera was proposed [6]. This scheme computes for each captured image a standard digital signature, and then the digital signature is stored and transmitted along with the image. Recently, Yeung and Mintzer [7] propose using an authentication watermark to protect the integrity of images. Yeung and Mintzer's authentication watermark uses a pseudo random sequence and a modified error diffusion method to embed a binary watermark to an image, so that any change in pixel values to the image can be detected. The pseudo random number generator is seeded using the key of the owner, hence associating the image (and the watermark) with the original owner.

In this paper we propose a watermark that allows a user with an appropriate security key to verify the integrity and the ownership of an image. Using the correct key, we can extract a watermark from a watermarked image that can be identified to be associated with an owner. Otherwise, if the user performs the watermark extraction procedure using either an incorrect key or with an image that was not watermarked, the user obtains an image that resembles random noise. Furthermore, this authentication watermark can detect and localize any change to the image, including changes in pixel values or image size.

The security of the watermarking algorithm relies on the computational infeasibility to break a cryptographic hash function. As a result, the security of the system

resides in the secrecy of the user key and not in the obscurity of the algorithm. In fact, the watermark insertion and extraction steps can be make public without compromising the security of the watermark.

From the perspective of an image viewer, watermarks can be classified into two categories: visible and invisible. Visible watermark refers to the type of technique where a visible stamp, e.g., a company logo, is inserted to the image [8]. The stamp is visible in similar fashion as the watermark in our dollar bills. There are generally two problem associated with visible watermarks. First a visible watermark must be difficult to be removed. In this regard, Braudaway *et al.* suggest the insertion of random noise to the watermarked image to increase the difficultly in manually removing the watermark [8]. Second, a visible watermark must be able to withstand the impersonator problem. It is easy for person A to insert the logo of B to an image and then claim that the resulting image came from B, while in fact B may not want to be associated with such an image. As a result, a visible watermark bearing a certain logo does not constitute a proof of ownership. We describe in this paper a secure visible watermarking method, where we use the security of an invisible watermarking algorithm to protect the visible watermark.

Invisible Authentication Watermark

We describe here the details of our authentication watermarking algorithm for grayscale images. For color images, the same technique can be applied independently to the color planes of the image, either in the RGB color space or in any other color space such as YUV.

Consider an image $x_{m,n}$ of size M by N pixels that we want to insert a watermark to form the watermarked image $y_{m,n}$ of the same size. We partition $x_{m,n}$ into blocks of $I \times J$ pixels. Our scheme inserts a watermark to each block of image data. The watermark insertion procedure for each image block is shown in Fig. 1.

Let $a_{m,n}$ be a bi-level image that we will use as our watermark, to be embedded in $x_{m,n}$. Note that $a_{m,n}$ needs not be of the same size as $x_{m,n}$. From $a_{m,n}$, we form another bi-level image $b_{m,n}$ of size $M \times N$ (same size as $x_{m,n}$). There are many ways of doing so. In our example, we form $b_{m,n}$ by tiling $a_{m,n}$, i.e., periodically replicating $a_{m,n}$ to the desired size. Another possibility is to append all zeros (or all ones) to the boundary of

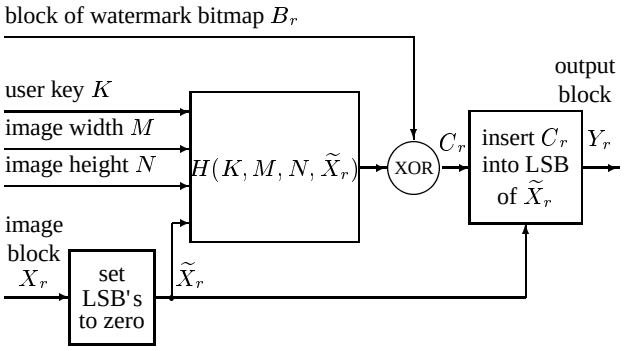

Figure 1: Watermark insertion procedure for each image block. The same watermark insertion procedure is applied independently to each block of the image.

$a_{m,n}$ so that we obtain $b_{m,n}$ of the desired size.

Let

$$X_r = \{x_{iI+k,jJ+l} : 0 \le k \le I-1; 0 \le l \le J-1\}$$

be a block of size $I \times J$ taken from the image $x_{m,n}$. For simplicity, we are using a single index r to denote the r^{th} block in the image. The corresponding block within the binary image $b_{m,n}$ is denoted

$$B_r = \{b_{iI+k,jJ+l} : 0 \le k \le I-1; 0 \le l \le J-1\}.$$

Consider a cryptographic hash function

$$H(S) = (d_1, d_2, \ldots, d_p)$$

where S represents a string of data of arbitrary length, d_i's are the binary output bits of the hash function, and p is the size of the output bit string. It has the property that given an input bit string S and its corresponding output $(d_1, \ldots, d_p)$, it is computationally infeasible to find another input bit string of any length that will be hashed to the same output $(d_1, \ldots, d_p)$. An example is the well known MD5 [9] where any string of data is hashed into a bit array of length 128, i.e., $p = 128$. For the rest of this paper, we will use MD5 as our hash function. It is obvious that any other cryptographic hash function can be used in our watermarking scheme. In our case, we need to choose the block size parameters I and J so that they satisfy $IJ \le p$.

Let K be a user key consisting of a string of bits. For each block of data X_r, we form the corresponding block $\widetilde{X}_r$ where each element in $\widetilde{X}_r$ equals the corresponding

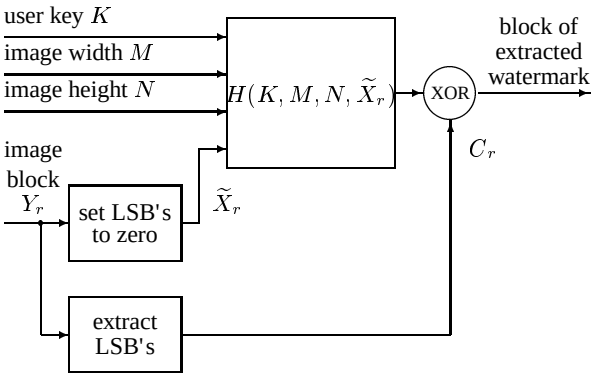

Figure 2: Block diagram of watermark extraction procedure for each image block.

element in X_r except that the least significant bit is set to zero. We compute for each block the hash

$$H(K, M, N, \widetilde{X}_r) = (d_1^r, d_2^r, \ldots, d_p^r).$$

Then, we select the first IJ bits in the hash output and form the rectangular array $d_{m,n}$ of size $I \times J$. This array is combined with B_r to form a new binary block C_r using a pixel by pixel exclusive-OR (XOR) operation. That is, we form

$$c_{m,n} = b_{m,n} \oplus d_i,$$

where $\oplus$ is the XOR operation, and $c_{m,n}$ are the elements in C_r. Finally we put $c_{m,n}$ into the least significant bit of the block $\widetilde{X}_r$ to form the output block Y_r. This procedure is repeated for each block of data, and all the output blocks Y_r are assembled together to form the watermarked image $y_{m,n}$.

The watermark extraction procedure is shown in Fig. 2. Note in particular that if we set the least significant bits of each element in the block Y_r to zero, we obtain a block $\widetilde{X}_r$ as the one in Fig. 1. For each block of data Y_r, we compute $H(K, M, N, \widetilde{X}_r)$ and perform a pixel by pixel XOR operation with C_r to form a block of the output binary watermark.

Properties of The Authentication Watermark

This watermarking approach allows the detection of any change to a marked image. The correct user key is required for the extraction of the proper watermark. The watermarking scheme exhibits the following properties:

Figure 3: An original image.

Figure 4: Watermarked image. This image should be visually identical to the original of Fig. 3.

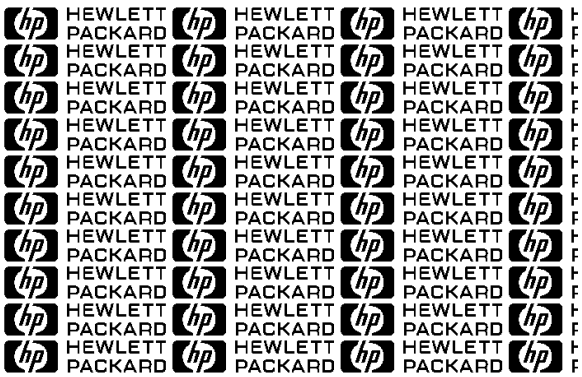

Figure 5: Extracted watermark from the image in Fig. 4.

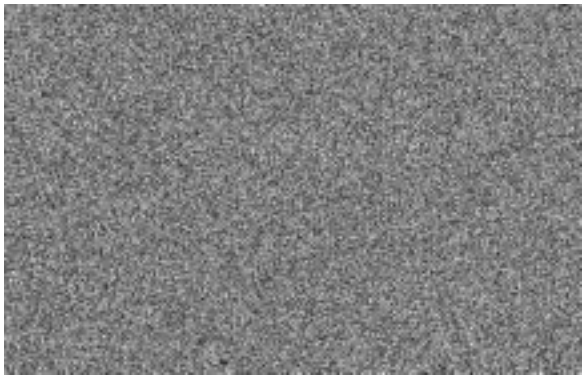

Figure 6: Extracted watermark from the image in Fig. 4 using an incorrect user key. The output resembles random noise. Similar output will result if the image contains no watermark, of if a watermarked image is cropped.

Figure 7: The watermarked image in Fig. 4 has been changed where a glass was pasted on top.

Figure 8: Extracted watermark from Fig. 7 indicating the location where changes have occurred.

- Fig. 3 shows an original image and Fig. 4 shows an image watermarked using the technique in this paper. It clearly demonstrates that the watermark is invisible.

- If one uses the correct user key K and applies the watermark extraction procedure to Fig. 4, one obtains an output image Fig. 5, indicating the presence of a proper watermark.

- If an image is unmarked, i.e., if it does not contain a watermark, the watermark extraction procedure returns an output that resembles random noise as shown in Fig. 6.

- If one applies an incorrect key (for example, if one does not know the key), then the watermark extraction procedure returns an output that resembles random noise as shown in Fig. 6.

- If a watermarked image is cropped and then one applies the watermark extraction procedure on this cropped image, the procedure returns an output that resembles random noise as shown in Fig. 6.

- If one changes certain pixels in the watermarked image, then the specific locations of the changes are reflected at the output of the watermark extraction procedure. Fig. 7 shows an image where a glass is pasted onto Fig. 4. Fig. 8 shows the extracted watermark from Fig. 7, indicating the specific area where changes have been made.

- A question that arises is that whether the watermark is secure if it is put into the least significant bit of the image. Recall that this watermark is designed for authentication purposes, i.e., to detect any change to the image. If someone attempts to remove the watermark by changing some bit planes of the image, the watermark extraction procedure will detect the changes.

- A very important issue is whether it is possible for someone to forge a watermark into the scheme. Consider an image block B_r. Suppose someone wants to alter some or all of the pixels in this image block so that it becomes $\widehat{B}_r$. It is necessary that the pixel values in the two image blocks satisfy

$$H(K, M, N, B_r) = H(K, M, N, \widehat{B}_r).$$

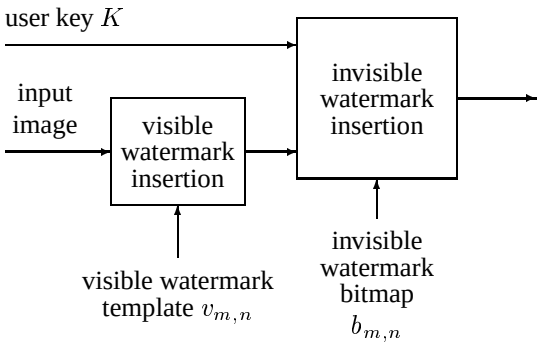

Figure 9: A method in using an invisible authentication watermark to protect a visible watermark.

That is, the digest generated from both image blocks must be identical. This is considered computationally infeasible because of the properties of cryptographic hash functions such as MD5 [9].

Secure Visible Watermark

As mentioned in the Introduction Section, we like to protect a visible watermark with an invisible authentication watermark, so that a person cannot insert a visible logo that belongs to somebody else. To this end, we first insert a visible watermark by modulating the pixel values according to the logo. We then insert the authentication watermark as described in an earlier section. The overall scheme is shown in Fig. 9. In such way, all the security features of the invisible watermark described in the previous section will hold for the visible watermark as well.

Consider a graylevel image $u_{m,n}$, where a grayscale visible watermark $v_{m,n}$ is to be inserted. As done previously, we assume that $v_{m,n}$ is of the same size as $u_{m,n}$. If this is not the case, we can always extend the watermark template in some fashion so that $v_{m,n}$ and $u_{m,n}$ are of the same size. The insertion of $v_{m,n}$ to $u_{m,n}$ is performed by amplitude modulating the pixels in $u_{m,n}$.

To insert the visible watermark, we first choose a parameter λ that controls the intensity of the visible watermark. That is, it controls how dark the visible watermark is on the output image. We then normalize the watermark template $v_{m,n}$ so that all the pixels fall within the range $\{0, 1, \ldots, \lambda\}$, and that the values 0 and λ are taken up by pixels within $v_{m,n}$. The insertion procedure for the visible watermark is simply

$$w_{m,n} = u_{m,n} - \lambda + v_{m,n}.$$

Figure 10: Image of Fig. 3 with both visible and invisible watermarks inserted. Here we used $\lambda = 30$. We can darken the visible watermark by using a larger λ. If we apply the invisible watermark extraction procedure to this image, we will obtain an image identical to Fig. 5.

This form ensures that a white pixel in $v_{m,n}$ will not change the corresponding intensity of the image $u_{m,n}$, while a black pixel in $v_{m,n}$ will darken the corresponding pixel $u_{m,n}$ by λ.

This insertion procedure can also be applied if $u_{m,n}$ is a color image. In such case, we simply modulate the luminance component of the color image. Consider the transformation between YUV and RGB color spaces (See, for example, [10].)

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.14 \\ 1 & -0.396 & -0.581 \\ 1 & 2.029 & 0 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix}.$$

In this case, modulating the luminance component Y is equivalent to independently modulating the R, G and B components. Similar observation holds for other color spaces such as YIQ and YCrCb.

Fig. 10 shows an image where both visible and invisible watermarks are added where $\lambda = 30$. If we want a darker visible watermark, we can increase the value of λ. Note that the template for the visible and invisible watermarks are complete independently of each other. If we apply to this image the invisible watermark extraction procedure as shown in Fig. 2, we will obtain an output image identical to Fig. 5.

References

[1] N. Memon and P. W. Wong, "Protecting digital me-

dia content: Watermarks for copyrighting and authentication," *Communications of ACM*. To appear.

[2] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent robust image watermarking," in *Proceedings of ICIP*, (Lausanne, Switzerland), pp. III 211–214, September 1996.

[3] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," Tech. Rep. 95-10, NEC Research Institute, 1995.

[4] N. Nikolaidis and I. Pitas, "Copyright protection of images using robust digital signatures," in *Proceedings of ICASSP*, May 1996.

[5] R. B. Wolfgang and E. J. Delp, "A watermarking technique for digital imagery: Further studies," in *Proceedings of International Conference on Imaging Science, Systems, and Technology*, (Las Vegas NV), 1997.

[6] G. L. Friedman, "The trustworthy digital camera: restoring credibility to the photographic image," *IEEE Transactions on Consumer Electronics*, vol. 39, pp. 905–910, November 1993.

[7] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proceedings of ICIP*, (Santa Barbara, CA), October 1997.

[8] G. W. Braudaway, K. A. Magerlein, and F. C. Mintzer, "Color correct digital watermarking of images." United States Patent 5530759, June 1996.

[9] R. L. Rivest, "The MD5 message digest algorithm." Internet RFC 1321, April 1992.

[10] R. W. G. Hunt, *The Reproduction of Colour in Photography, Printing & Television*. England, UK: Fountain Press, fourth ed., 1987.