HIERARCHICAL MULTIPLE IMAGE WATERMARKING FOR IMAGE AUTHENTICATION AND OWNERSHIP VERIFICATION

Jagdish C. Patra, Kah K. Ang and Ee-Luang Ang

School of Computer Engineering
Nanyang Technological University
Singapore
Email: {aspatra@ntu.edu.sg, hybrid79@yahoo.com and aselang@ntu.edu.sg}

ABSTRACT

We propose a novel scheme in which two black and white images are embedded as watermarks into a host image. The first watermark is intended for secure image authentication and the second one is to verify the ownership. The two watermarks are embedded into the least significant bits of the host image using a private secret key in such a way that these images remain perceptually invisible to human eyes. The multiple watermark images can be easily extracted at the receiver site blindly, i.e., without aid of the original host image. If the watermarked image is tampered in any way, the recipient can only extract the first watermark image and the region of tampering will appear on it, whereas the second extracted watermark image will reveal only a random noisy image.

1. INTRODUCTION

In the present digital world, digital images and documents can be easily duplicated, modified, transformed, and diffused by readily available, powerful image processing softwares. It is usually impossible to differentiate the original image from the manipulated image by visual inspection. Further, the explosive growth of the Internet and its massive use in distribution of huge amount of multimedia content give rise to the important issues of image authentication and ownership verification. Image authentication is of prime importance in e-commerce, legal applications, medical data archiving and news reporting. It can be viewed as a process to ensure that the digital image in question truly reflects what the scene looked like at the time of its capture or creation. Ownership verification is another issue in which it is to be confirmed that the digital image indeed belongs to the rightful owner.

One of the earliest techniques for image authentication is modifying the contents of the least significant bit (LSB) of a pixel value in the host image [1]-[3]. However, as the inserted watermark influences only the LSBs of the host image pixels, the embedded watermark can easily be extracted, thus a fake watermark can be added to another manipulated image. To overcome this problem, Wong and Menon [4] proposed an elegant and effective image authentication and ownership verification scheme. The LSB of each pixel of the host image is used to embed a black and white (b/w) image using MD5 hash function and XOR operations. The scheme can authenticate the received image by extracting the hidden b/w image using the correct secret key. In case of any tampering of the watermarked image, the region of tampering (RoT) can be visible on the extracted watermark image. In this scheme, only one watermark image is embedded into the host image for the purpose of both image authentication and ownership verification. A comprehensive survey and some of the recently proposed effective image authentication techniques may be found in [6]-[8].

In this paper, we propose a secure fragile watermarking scheme for image authentication and ownership verification by embedding two b/w watermark images hierarchically into the host image using a secret key and the MD5 hash function. The major difference between our scheme to that of [4] is that we embed one image for image authentication and another for ownership verification. Besides, more security is provided to the second watermark as we introduce the hash value of the first watermark during insertion of second watermark. The two watermark images can be extracted easily at the receiver site blindly, i.e., without aid of the original host image. If no tampering is made to the watermarked image, then using the secret key, the recipient can extract the first watermark image which authenticates the host image. The second watermark to verify the ownership can be extracted successfully only if the first watermark is extracted correctly.

However, if the watermarked image is tampered, then the recipient can extract only the first watermark image and the RoT will appear on it. If he attempts to extract the second watermark, the extracted image will reveal only a meaningless random noisy image.

It may be noted that as this scheme is intended mainly for image authentication, it will fail if compression and/or any signal processing is made to the watermarked image or document.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

2. WATERMARKING OF TWO B/W WATERMARKS

Two watermarks, each consisting of a b/w image, are embedded hierarchically into the host image in such a way that the binary images remain invisible and distortion introduced in the host image is quite low.

2.1. Watermark embedding

Let the host image, X be a 256 level gray scale image of size $M \times N$. Thus, a pixel, $x_{m,n}$ of X can be represented by a byte. Let $A1$ be a bi-level (b/w) image of size $I \times J$ ($I < M$ and $J < N$). By tiling (periodically replicating), $A1$ is converted into a image $W1$ of size $M \times N$. In a similar manner, another b/w image $W2$ of size $M \times N$ is generated from $A2$. As $W1$ and $W2$ are b/w images, their pixels $w_{m,n}$ can be encoded by a single bit. Two watermark images $W1$ and $W2$ will be embedded into the host image X to generate a watermarked image X^{W12}. The details of the embedding scheme using two passes are depicted in Fig. 1.

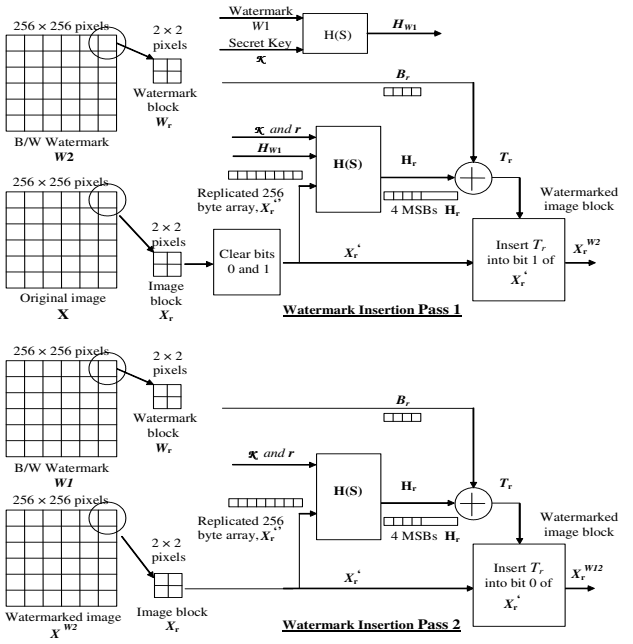

Fig. 1. Hierarchical watermark embedding process with M=N=256 and K=L=2.

Pass 1: Insertion of the second watermark

Step 1. Feed the first watermark W1 ($M \times N$ bits) and the secret key $\mathcal{K}$ to a MD5 hash function to produce a 128-bit hash output H_{W1}.
Step 2. Divide the host image X of size $M \times N$ into several non-overlapping blocks X_r, each of size $K \times L$ ($K, L <$ 128). Thus, there will be total of N_B blocks, where $N_B = (M \times N)/(K \times L)$, the block index $r = 1, 2, \cdots, N_B$, and each block consists of $K \times L$ bytes.
Step 3. Select a block r. Clear bits 0 and 1 from all the $K \times L$ bytes of X_r to form a modified block $X_r^{'}$.
Step 4. The $X_r^{'}$ is replicated several times to form another block $X_r^{"}$ of size 16×16. Thus, $X_r^{"}$ will have 256 bytes.
Step 5. The watermark image $W2$ of size $M \times N$ is also divided into non-overlapping N_B blocks each of size $K \times L$. Let the rth block be denoted by W_r. As the watermark is a b/w image, W_r will have $K \times L$ bits. Let these $K \times L$ bits be denoted by B_r.
Step 6. Feed the block index r, secret private key $\mathcal{K}$, the block $X_r^{"}$ and the hash output H_{W1} to a MD5 hash function to generate a 128-bit string denoted by H_r.
Step 7. Carry out exclusive OR (XOR) operation between the $K \times L$ MSBs of H_r and the $K \times L$ bits of B_r (obtained from Step 5). Let the XOR output be denoted by T_r.
Step 8. Insert the $K \times L$ bits of T_r into bit 1 of the image block $X_r^{'}$ to generate a block X_r^{W2}.
Step 9. Repeat steps 3 through 8 until all the blocks are completed. Finally, cascade all the N_B blocks of X_r^{W2} to form the image X^{W2} of size $M \times N$. This image is the host image embedded with watermark $W2$.

Pass 2: Insertion of the first watermark

Step 1. The watermarked image X^{W2} is divided into N_B blocks each of size $K \times L$. Let the rth block be denoted by X_r.

Step 2. Select a block r. As the bit 0 is already cleared during Pass 1, let us denote X_r by $X_r^{'}$. Replicate $X_r^{'}$ few times to form another image $X_r^{"}$ of size 16×16.

Step 3. Divide the b/w watermark image $W1$ into N_B blocks each of size $K \times L$. Let the rth block be denoted by W_r and the $K \times L$ bits of W_r by B_r.

Step 4. Feed the block index r, secret key $\mathcal{K}$, and the block $X_r^{"}$ to MD5 hash function. Let the 128-bit hash output be denoted by H_r.

Step 5. Carry out XOR operation between the $K \times L$ MSBs of H_r and B_r (obtained from Step 3). Let the $K \times L$ bits of XOR output be denoted by T_r.

Step 6. Insert the $K \times L$ bits of T_r into bit 0 of the block $X_r^{'}$ to form a block denoted by X_r^{W12}.

Step 7. Repeat steps 2 through 6 to complete all the blocks and then cascade the watermarked blocks together to form an image denoted by X^{W12} of size $M \times N$. This image represents the host image X in which the two watermarks $W1$ and $W2$ have been embedded.

2.2. Watermark extraction

Extraction of watermarks is a reverse process of embedding and is shown in Fig. 2. The two watermarks $W1$ and $W2$ can be extracted from the received watermarked image Y^W with two passes using the following steps.

Pass 1: Extraction of the first watermark

Step 1. Divide the received watermarked image Y^W into N_B non-overlapping blocks each of size $K \times L$.

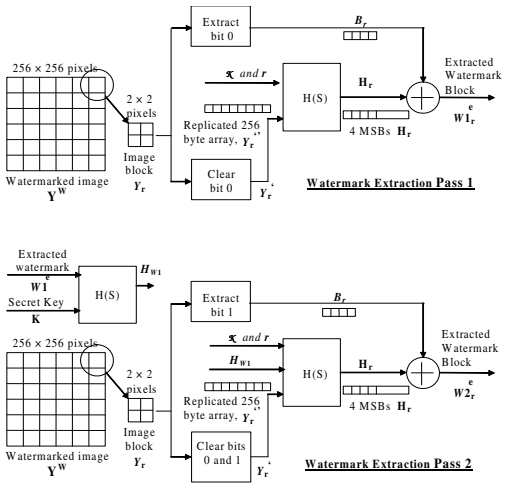

Fig. 2. Hierarchical watermark extraction process with M=N=256 and K=L=2.

Step 2. Select a block r and denote it by Y_r. Extract bit 0 from all the $K \times L$ bytes of Y_r to form a $(K \times L)$-bit string. Let this string be denoted by B_r.

Step 3. Clear bit 0 from all the $K \times L$ bytes of Y_r to form a block $Y_r^{'}$. Next, replicate $Y_r^{'}$ blocks to form another block $Y_r^{"}$ of size 16×16.

Step 4. Feed the block index r, the secret key $\mathcal{K}$, and the image block $Y_r^{"}$ to a MD5 hash function to generate a 128-bit string. Let this bit string be denoted as H_r.

Step 5. Carry out XOR operation between the $K \times L$ MSBs of H_r and B_r (obtained from Step 2) to produce a bit string denoted by $W1_r^e$.

Step 6. Repeat the steps 2 through 5 until all N_B blocks of watermark are extracted. Then, cascade all the blocks $W1_r^e$, $r = 1, 2, \cdots, N_B$ to generate the extracted watermark $W1^e$ of size $M \times N$.

Pass 2: Extraction of the second watermark

Step 1. Feed the secret key $\mathcal{K}$ and the extracted first watermark $W1^e$ to a MD5 hash function. Let the 128-bit hash output string be denoted by H_{W1}.

Step 2. Divide the received watermarked image Y^W into N_B non-overlapping blocks of size $K \times L$. Let the rth block be denoted by Y_r.

Step 3. Select a block Y_r and extract the bit 1 from all the bytes of the block to form a $(K \times L)$-bit string. Let this string be denoted by B_r.

Step 4. Clear bit 1 and bit 0 from all the $K \times L$ bytes of Y_r to form a block denoted by $Y_r^{'}$. Next, replicate $Y_r^{'}$ few times to form an expanded block of size 16×16 and denote it by $Y_r^{"}$.

Step 5. Feed the secret key $\mathcal{K}$, block index r, hash output H_{W1} and $Y_r^{"}$ to a MD5 hash function to generate 128-bit string. Let this string be denoted as H_r.

Step 6. Carry out XOR operation between the $K \times L$ MSBs of H_r and B_r (obtained from Step 3). Let $W2_r^e$ denotes the $(K \times L)$-bit string obtained from the XOR operation.

Step 7. Repeat steps 3 through 6 until all N_B watermark blocks are extracted. Then, cascade the blocks $W2_r^e$, $r = 1, 2, \cdots, N_B$, to generate the extracted watermark $W2^e$ of size $M \times N$.

2.3. Image authentication and ownership verification

One can determine the authenticity of the image and verify the ownership by visual inspection of the two extracted b/w images $W1^e$ and $W2^e$. The first watermark is intended for image authentication, i.e., to verify whether the watermarked image has been tampered or not. Whereas, the second watermark is meant to verify the owner of the host image.

If the correct secret key $\mathcal{K}$ is used during the extraction process, the first watermark $W1^e$ will emerge into a meaningful image. Further, if the watermarked image has been tampered, then the RoT will appear on it. However, as the image has been tampered, the extracted second watermark will result in meaningless random noise. The extracted second watermark will appear as a meaningful b/w image only if the image has not been tampered with.

3. SIMULATION AND RESULTS

The original Lenna image (256×256) and the two b/w watermark images used in the simulated experiments are shown in Fig. 3. A block size of 2×2 ($K = L = 2$) was chosen for the embedding process. The first and second watermark images were embedded into the host Lenna image using the above scheme. The watermarked image is shown in Fig. 4(a). By a simple visual inspection, one can not differentiate between the original (Fig. 3(a)) and the watermarked image (Fig. 4(a)).

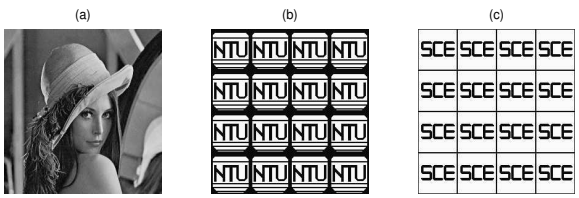

Fig. 3. (a) The original Lenna image. (b) The first b/w watermark image. (c) The second b/w watermark image.

The MSE between the watermarked (Fig. 4(a)) and original images (Fig. 3(a)) was found to be 4.02 dB (PSNR=44.1 dB). Watermark extraction was attempt-ed using a wrong secret key and its results are shown in Fig. 4(b) and (c). The extracted images correspond to meaningless random noise. This shows that image authentication can be made only by authorized persons possessing the secret key.

The results of image authentication using the correct secret key $\mathcal{K}$ is shown in Fig. 5. The received watermarked image has been tampered at two places (upper right and center). Watermark extraction from this tampered image was

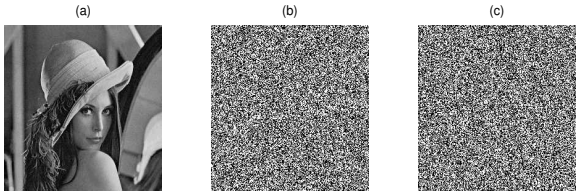

Fig. 4. Watermark extraction using a wrong secret key. (a) The watermarked Lenna image. (b) Extracted first watermark image. (c) Extracted second watermark image.

carried out using the correct secret key and the results are shown in Fig. 5 (b) and (c). The first extracted watermark (W1) shown in Fig. 5(b) clearly indicates the two regions of tampering. However, as the watermarked image was tampered, the extracted second watermark image (Fig. 5(c)) generated a meaningless random noisy image.

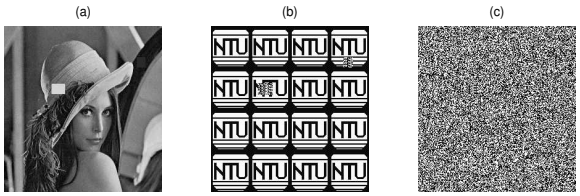

Fig. 5. Watermark extraction using the correct secret key. (a) Tampered watermarked Lenna image. (b) Extracted first watermark image. (c) Extracted second watermark image.

Next, extraction process was carried out from an untampered watermarked image using the correct secret key. The watermarked host image, extracted first and second watermarks are shown in Fig. 6 (a), (b) and (c), respectively. Successful extraction of the first watermark indicates positive image authentication and extraction of the second watermark successfully verifies the ownership. Thus, one can determine authenticity and verify ownership of the image by extracting the hidden images (watermarks) from the received image.

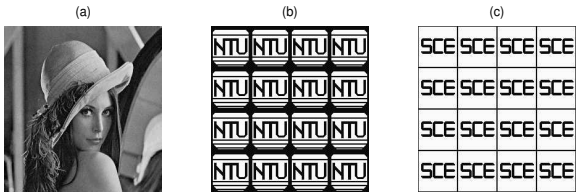

Fig. 6. Watermark extraction using the correct secret key. (a) Untampered watermarked Lenna image. (b) Extracted first watermark image. (c) Extracted second watermark image.

4. CONCLUSIONS AND DISCUSSIONS

We have proposed a hierarchical multiple image watermarking scheme for digital images. The main purpose of embedding two watermarks is that each watermark can be used for a specific purpose. In this scheme, the first and second watermarks are used for image authentication and ownership verification, respectively.

Two b/w watermark images are inserted into two LSBs of the host image in such a way that the watermark images remain perceptually invisible. The recipient can verify the authenticity of the image and the ownership by extracting the first and second watermarks, respectively, with the correct secret key. If the received image is untampered, then both extracted watermarks will lead to meaningful images. On the other hand, if the received image has been tampered, then the first watermark will show the region of tampering on it, but the second watermark will show a meaningless random noise. This indicates that the ownership verification can be carried out only with an untampered image. Further, if one attempts to extract the watermark(s) using a wrong secret key, then the watermark(s) will lead to meaningless random noise.

5. REFERENCES

[1] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. IEEE Intl. Conf. Image Processing*, vol. 2, Austin, TX, 1994, pp. 86-90.

[2] S. Walton, "Information authentication for a slippery new age," *Dr. Dobbs Jnl.*, vol. 20, pp. 18-26, Apr. 1995.

[3] R. B. Wolfgang, and E. J. Delp, "A watermark for digital images," in *Proc. IEEE Intl. Conf. Image Processing*, vol. 3, 1996, pp. 219-222.

[4] P. W. Wong and N. Menon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Tran. Image Processing*, vol. 10, no. 10, pp. 1593-1601, Oct. 2001.

[5] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proc. IEEE ICIP*, Santa Barbara, CA, Oct. 1997, vol. 2, pp. 680-683.

[6] C-Y. Lin and S-F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits and Systems of Video Technology*, vol. 11, no. 2, , pp. 153-168, Feb. 2001.

[7] C. Rey and J-L. Dugelay, "A survey of watermarking algorithms for image authentication," *EURASIP Applied Signal Processing*, vol. 6, pp. 613-621, 2002.

[8] 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, pp. 585-595, June 2002.