Security of fragile watermarking scheme for image authentication

K-C Liao[a], **W-B Lee***[a] and **C-W Liao**[b]
[a]Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan

[b]Department of Information Management, Hsiuping Institute of Technology, Taichung, Taiwan

Abstract: Recently, Lu *et al*. proposed a fragile watermarking scheme to prevent the quantization attack. However, the present study shows that the modified scheme's objective as a fragile watermark is unfortunately lost, because it is possible for any attacker to copy a watermark from one image into another without knowing the watermark owner's insertion keys.

Keywords: fragile watermarking, image authentication

1 INTRODUCTION

The rapidly growing multimedia market and use of digital images in general have revealed an urgent need for securing digital images. The digital watermarking technique, which allows the information to be embedded into a multimedia content, has emerged as a widely accepted approach for identification or authentication purposes.

Digital watermarking schemes can be divided into two categories: (1) the robust watermark, which may find application in copyright protection, since it is generally designed to withstand malicious attacks such as scaling, cropping and compression; (2) the fragile watermark, which is useful for purposes of image authentication, and can potentially be used to verify the integrity of a given watermarked image.

Some block-based fragile watermark schemes[1,2] for image authentication and verification have been proposed, but these schemes are vulnerable to quantization attack, as pointed out by Holliman and Memon[3]. Recently, Lu *et al*.[4] proposed a pixel-wise fragile watermark scheme, which claimed to be able to withstand quantization attack and locate the exact tampered regions on the watermarked image.

Unfortunately, Lu *et al*.'s modified scheme is still impractical, because it is potentially vulnerable to attack, whereby counterfeit watermarks can be inserted into images without the consent of original watermark owner. In such a way, a malicious broadcaster might attempt to insert a watermark into any other source material, and charge for a commercial transmission. It implies that Lu *et al*.'s modified version would clearly be insufficient to satisfy the authentication requirements of the fragile watermarking technique.

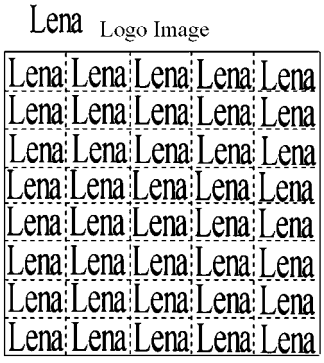
2 IMAGE AUTHENTICATION SCHEME

In Lu *et al*.'s scheme, an 8-bit greyscale image I of size $M \times N$ is given, and a binary watermark image W of the same size is also assumed. In practice, an $M \times N$ binary watermark image is usually constructed by tiling the smaller logo image to the desired size, as illustrated in Fig. 1.

2.1 Watermark insertion procedures

The watermark W is embedded by carrying out the following procedures, and the steps are summarized in Fig. 2.

The MS was accepted for publication on 30 January 2006.
* *Corresponding author: Prof. Wei-Bin Lee, Department of Information Engineering and Computer Science, Feng Chia University, 100, Wenhwa Road, Seatwen, Taichung 407, Taiwan; e-mail: wblee@fcu.edu.tw*

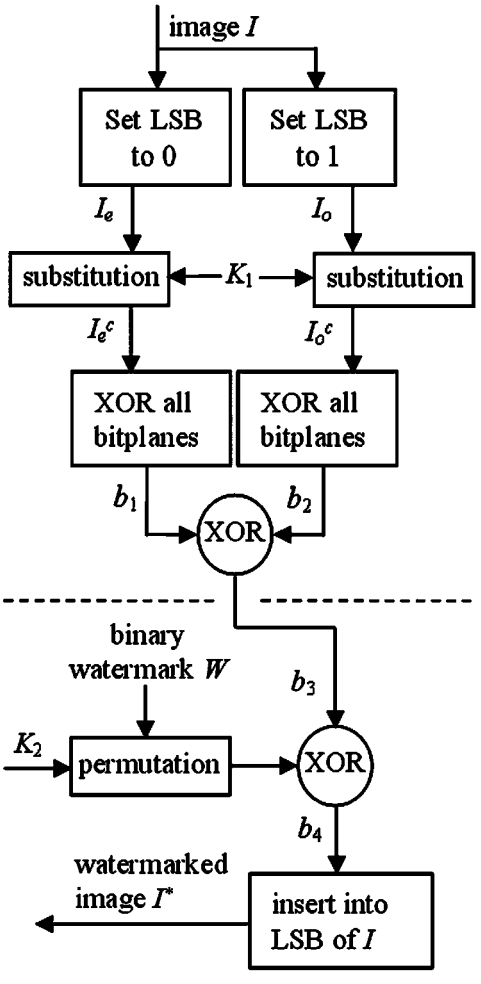


1 Tiling of logo image

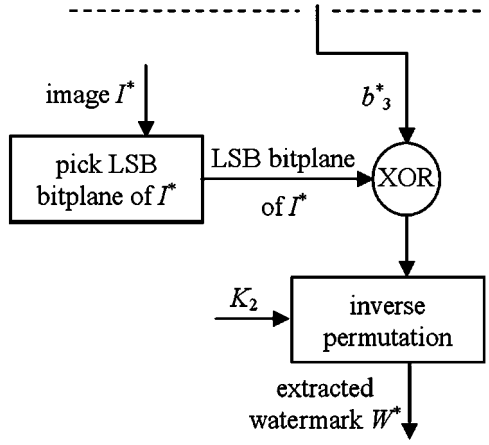
Step 1. Create L_p by permuting the intensity vector $L=[0, 1, ..., 255]$ with a secret key K_1.

Step 2. Produce two images I_e and I_o from I by settingthe least significant bit (LSB) bit plane to 0 and 1, respectively.

Step 3. Create two images I_e^c and I_o^c, using L_p as a look-up-table to encipher I_e and I_o, respectively, by $I_e^c(i, j)=L_p[I_e(i, j)]$, $I_e^c(i, j)=L_p[I_e(i, j)]$, where (i, j) denotes the position of the ith row and jth column.



2 Block diagram for watermark embedding



3 Block diagram for watermark extracting

Step 4. Obtain two binary images b_1 and b_2 by making exclusive OR operations on the 8-bit plane of I_e^c and I_o^c, respectively.

Step 5. XOR b_1 and b_2 pixel by pixel to obtain another binary image b_3, where XOR is an exclusive OR operation.

Step 6. Use the second secret key K_2 to generate a permutation vector and permute the pixel positions of the watermark W to form a permuted watermark W_p.

Step 7. XOR W_p and b_3 pixel by pixel to obtain a new binary image b_4.

Step 8. Replace the LSB bit plane of the host image I by b_4 to form the watermarked image I^*

2.2 Watermark extraction procedures

For an input image I^*, the watermark extraction procedures are carried out as follows, and the block diagram is shown in Fig. 3.

Step 1. Obtain a binary image b_3^* by performing the same steps as Steps 1–5 in the embedding stage.

Step 2. XOR b_3^* and the LSB bit plane of I^* pixel by pixel to generate a binary image b_4^*.

Step 3. Permute b_4^* inversely with the secret key K_2 to extract the watermark W^*.

3 POSSIBLE ATTACK ON SCHEME

The security of Lu *et al.*'s scheme lies in the secret keys K_1 and K_2, where K_1 is used for pixel

4 (a) Host image 'Lena'; (b) watermark image; (c) watermarked image; (d) extracted watermark, NHS=1.0

substitution encryption, and K_2 for permutation of watermark. Without loss of generality, assume that an image I^* contain an owner's watermark W inserted using secret keys K_1 and K_2.

However, given watermarked I^* and an unwatermarked image Y, it is possible for an attacker to insert a watermark into image Y without the knowledge of K_1 and K_2. Therefore, it is possible to forge authenticated images without the consent of the original watermark owner, and the details are described as follows.

Step 1. Classify the pixels in I^* to 128 group: $\{G_0, G_1, G_2, \ldots, G_{127}\}$ according to the seven most significant bits (MSB) of each pixel.

Step 2. Construct a binary table $T(G_i)=V_i$, where $V_i \in \{0,1\}$ (for $i=0,1,2,\ldots,127$) is determined by the majority of the LSB value of the pixels in G_i. (Here, the size of look-up table is 128, and V_i records the LSB value that occurs most frequently within each group G_i. For instance, if there are n elements within group G_j, and over $n/2$ of their LSBs is 1, then V_j will equal 1.)

Step 3. Obtain a binary image b'_3 by inputting a 7-bit greyscale image I_7^* and using the $T(G_i)$ as a look-up table, where I_7^* is I^*, but ignore the LSB of each pixel in I^*.

Step 4. Create another binary image $\overline{b'_3}$ by complementing each pixel value of b'_3.

Step 5. XOR $\overline{b'_3}$ and the LSB bit plane of I^* pixel by pixel to generate a binary image W^*, where XOR is an exclusive OR operation.

Step 6. Obtain a binary image b_Y by inputting a 7-bit greyscale image Y_7 and using the $T(G_i)$ as a look-up table, where Y_7 is Y, but ignore the LSB of each pixel in Y.

Step 7. Create another binary image $\overline{b_Y}$ by complemening each pixel value of b_Y.

Step 8. XOR W^* and $\overline{b_Y}$ pixel by pixel to obtain a new binary image b'_4.

Step 9. Replace the LSB bit plane of the image Y by b'_4 to form the forged watermarked image Y^*.

4 EXPERIMENTAL RESULTS

Some experiments are conducted to demonstrate the effectiveness of the attack. Figue 4a, b shows the host image and the watermark image, respectively. After performing Lu *et al.*'s fragile watermarking scheme by using two keys K_1 and K_2, the watermarked image is shown in Fig. 4c, and the extracted result is shown in Fig. 4d. The present paper employs the normalized hamming similarity (NHS) to evaluate the effectiveness of the proposed attacks. Let W be the embedded binary watermark of size $A_w \times B_w$ and W^* be the extracted one with the same size. The NHS between W and W^* is defined as

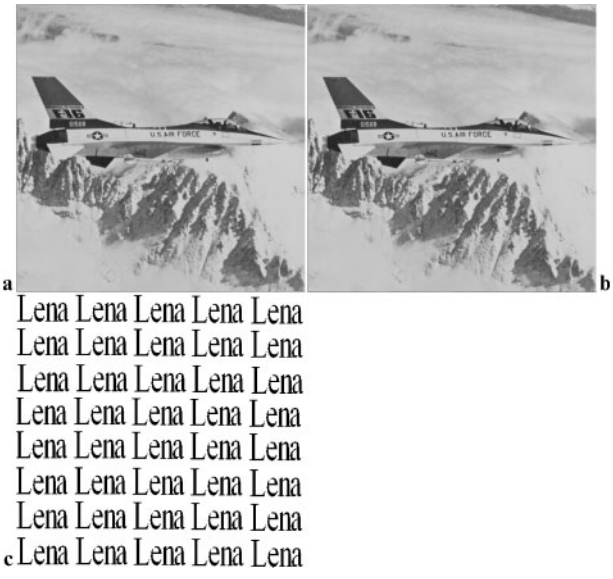
$$\mathrm{NHS} = 1 - \frac{\mathrm{HD}(W, W^*)}{A_w \times B_w}$$

where $\mathrm{HD}(\cdot,\cdot)$ denotes the number of bits different in the two binary images. It is not difficult to understand that NHS $\in$ [0, 1]. And the higher the NHS values acquired, the more similar the extracted watermark is to the embedded one.

From the viewpoint of different scenarios, the attack can be divided into two types.:

Type 1. Modification attack. In this attack, an attacker can perform a malicious modification in an authenticated image without being detected. Given the watermarked image (Fig. 4c) and a tampered watermarked image (Fig. 5a), the attacker can execute the proposed attack to derive the forged watermarked image, and the result is also shown in Fig. 5b. Then, the extractor can recover the genuine watermark (Fig. 5c) with NHS=1.0 from the forged watermarked image (Fig. 5b). This

5 (a) Modified watermarked image; (b) forged watermarked image; (c) extracted watermark, NHS=1.0



6 (a) Unwatermarked image; (b) forged watermarked image; (c) extracted watermark, NHS=1.0

implies that the attacker has successfully modified the watermarked image without being detected.

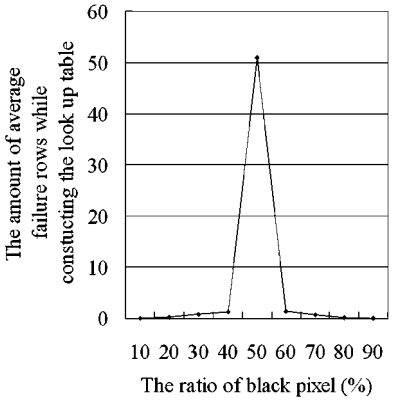
Type 2. Counterfeit attack. In this attack, an attacker can insert the watermark into another unwatermarked image. Given the watermarked image (Fig. 4c) and an unwatermarked image (Fig. 6a), the attacker can execute the proposed attack to derive the valid watermarked image shown in Fig. 6b, because the extractor can recover the genuine watermark (Fig. 6c) with NHS=1.0 from Fig. 6b. Thus, this implies that the attacker has successfully impersonated the watermark owner to forge the validity of the watermarked image.

Base on the above experimental results, it is not difficult to see that Lu *et al.*'s fragile watermarking scheme not only fails to achieve tamper detection and localization, but also violates the authentication purpose of the fragile watermarking scheme.

5 DISCUSSION

Observing the watermark image (Fig. 4b) used in the present experiment, the number of white pixels is obviously greater than the number of black ones. This property gives the attacker some statistical information to construct the look-up table $T(G_i)$. It is also common to observe that the number of black and white pixels in most binary watermark logs are not absolutely balanced, as in the logos of some well-known companies, such as IBM, Nike and Sony. It is not difficult to realize that the attack is so practical because the look-up table can be constructed successfully and be utilized to forge the watermark arbitrarily.

Here, an experiment is further given to demonstrate that the probability of successfully constructing the look-up table depends on the ratio of the number of black pixels in the watermark image. Since the pixels in the binary watermark image will be permutated before embedding and owing to the difficulty of finding images with the exact ratio of black pixels as needed, 10 binary image samples are generated randomly with different ratios of black pixel to facilitate the experiment. Figure 7 illustrates



7 Error probability for different ratio of black pixel

the number of average failure rows while constructing the look-up table (128 rows) using binary image samples with different ratios of black pixels as the permutated watermark input. This shows that the larger the difference between the numbers of black and white pixels, the higher the probability of the attack being successfully. Owing to fact that the numbers of black and white pixels in most binary watermark logs are not absolutely balanced, there is a high probability of constructing the look-up table successfully to fool the verifier. It also implies that Lu *et al*'s scheme is impractical.

6 CONCLUSION

It has been shown that Lu *et al.*'s fragile watermarking scheme is impractical, because it is potentially vulnerable to attack whereby counterfeit watermarks can be inserted into images without the consent of the original watermark owner.

REFERENCES

1 Wong, P. W. A watermark for image integrity and ownership verification, Proc. IS&T PIC Conf.: *IS&T 1998*, Portland, USA, May 1998, pp. 374–379.

2 Wong, P. W. and Memon, N. Secret and public key image watermarking schemes for image authentication and ownership verification. *IEEE Trans. Image Process.*, 2002, **10**, 1593–1601.

3 Holliman, M. and Memon, N. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. *IEEE Trans. Image Process.*, 2000, **9**, 432–441.

4 Lu, H., Shen, R. and Chung, F. L. Fragile watermarking scheme for image authentication. *Electron. Lett.*, 2003, **39**, 898–900.