PUBLIC KEY WATERMARKING FOR REVERSIBLE IMAGE AUTHENTICATION

Sang-Kwang Lee[1]*, Young-Ho Suh*[1]*, and Yo-Sung Ho*[2]

[1]Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong-Dong, Yuseung-Gu, Daejeon 305-700, KOREA
[2]Gwangju Institute of Science and Technology (GIST)
1 Oryong-Dong, Buk-Gu, Gwangju 500-712, KOREA
{sklee, syh}@etri.re.kr, hoyo@gist.ac.kr

ABSTRACT

In this paper, we propose a new public key watermarking scheme for reversible image authentication where if the image is authentic, the distortion due to embedding can be completely removed from the watermarked image after the hidden data has been extracted. This technique utilizes histogram characteristics of the difference image and modifies pixel values slightly to embed more data than other lossless data hiding algorithm. We show that the lower bound of the PSNR(peak-signal-to-noise-ratio) values of watermarked images are 51.14 dB. Moreover, the proposed scheme is quite simple and the execution time is rather short. Experimental results demonstrate that the proposed scheme can detect any modifications of the watermarked image.

***Index Terms*—** Data security, image processing

1. INTRODUCTION

In many circumstances, alternations to content serve legitimate purposes. However, in other cases, the changes may be intentionally malicious or may inadvertently affect the interpretation of the content. For example, an inadvertent change to an X-ray image might result in a misdiagnosis, whereas malicious tampering of photographic evidence in a criminal trial can result in either a wrong conviction or acquittal. Thus, in applications for which we must be certain a content has not been altered, there is a need for verification or authentication of the integrity of the content [1].

In most authentication techniques based on watermarking, the original image is inevitably distorted due to the authentication itself. Typically, this distortion cannot be removed completely due to quantization, bit-replacement, or truncation at the grayscale 0 and 255. Although the distortion is often quite small, it may be unacceptable for medical or legal imagery or images with a high strategic importance in certain military applications. Thus, it is desired to undo the changes introduced by authentication if the image is verified as authentic. Data embedding techniques satisfying this requirement, are referred to as *reversible* image authentication techniques.

Fridrich *et al.* [2] proposed a reversible authentication technique for images based on lossless bit-plane compression. They used the lowest bit-plane that provides enough space for the image hash after lossless compression. The disadvantage of this method is that for some noisy images we may have to use higher bit-planes and the distortion due to authentication can become visible. Even though the distortion can be completely removed, it is clearly desirable to preserve as much of the visual content of the image as possible.

In the previous work [3], we proposed a secret key reversible watermarking scheme for authentication purposes. Being a secret key scheme, the same key is used for both watermark embedding and extraction. Hence, the key must be transmitted from the owner to the verifier through a secure channel. In this paper, we extend the secret key watermarking scheme so that the integrity and ownership of the image can be verified using a public key. In order to verify the integrity of the image, we used the MD5 [4] as our hash function and the RSA public key encryption algorithm [5] for encryption and decryption. The hash code is combined with a binary logo image by a bit-wise exclusive OR(XOR) and then embedded in the histogram of the difference image from the original image.

2. PROPOSED AUTHENTICATION SCHEME

2.1. Watermark Embedding

The watermark embedding procedure of the proposed authentication scheme is shown in Fig. 1.

For a grayscale image $I(i,j)$ of size $M \times N$ pixels, we form the difference image $D(i,j)$ of size $M \times \frac{N}{2}$ from the original image. For $0 \leq i \leq M-1$ and $0 \leq j \leq \frac{N}{2}-1$,

$$D(i,j) = I(i,2j+1) - I(i,2j) \tag{1}$$

where $I(i,2j+1)$ and $I(i,2j)$ are the odd-line field and the even-line field, respectively. For watermark embedding, we empty the histogram bins of -2 and 2 by shifting some pixel values in the difference image. If the difference value is

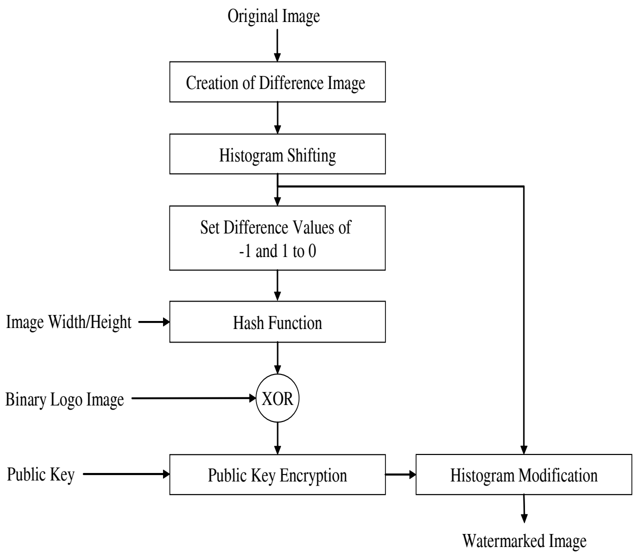


Fig. 1. Watermark embedding

greater than or equal to 2, we add one to the odd-line pixel. If the difference value is less than or equal to -2, we subtract one from the the odd-line pixel. Then, the modified difference image $\widetilde{D}(i,j)$ can be represented as

$$\widetilde{D}(i,j) = \widetilde{I}(i,2j+1) - I(i,2j) \tag{2}$$

where

$$\widetilde{I}(i,2j+1) = \begin{cases} I(i,2j+1)+1 & \text{if } D(i,j) \geq 2 \\ I(i,2j+1)-1 & \text{if } D(i,j) \leq -2 \\ I(i,2j+1) & \text{otherwise} \end{cases} \tag{3}$$

In this paper, we use MD5 as a hash function. The MD5 algorithm takes as input a message of arbitrary length and produces as output a bit array of 128. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest [4]. Let

$$\begin{aligned} \mathbf{D_s} &= \{D_s(i,j) : 0 \leq i \leq M-1; 0 \leq j \leq \tfrac{N}{2}-1\} \\ \widetilde{\mathbf{D}}_\mathbf{s} &= \{\widetilde{D}_s(i,j) : 0 \leq i \leq M-1; 0 \leq j \leq \tfrac{N}{2}-1\} \end{aligned} \tag{4}$$

Then, we compute the hash $A(l)$ of length 128 as follows:

$$H(M, N, \widetilde{\mathbf{D}}_\mathbf{s}) = \{A(l) : 0 \leq l \leq 127\} \tag{5}$$

where $\widetilde{\mathbf{D}}_\mathbf{s}$ equals the corresponding element in $\mathbf{D_s}$ except difference values of -1 and 1 are set to zero.

In order to generate a watermark $W(m,n)$ of the same size as embedding capacity C, we combine the hash $A(l)$ with the binary logo image $B(m,n)$ of size $P \times Q$ pixels using the bit-wise XOR operation. In the case that $C > 128$ and $C > P \times Q$, we form $\widetilde{A}(l)$ and $\widetilde{B}(m,n)$ by periodically replicating $A(l)$ and $B(m,n)$ to the length C, respectively. Then, the authentication watermark is computed as

$$W(m,n) = \widetilde{A}(l) \oplus \widetilde{B}(m,n) \tag{6}$$

where $\oplus$ is the bit-wise XOR operation. Then, we encrypt $W(m,n)$ with a public key cryptographic system to give

$$S(m,n) = Z_E(K_p, W(m,n)) \tag{7}$$

where $Z_E(\cdot)$ is the encryption function of the public key system and K_p is the public key.

In the histogram modification process, the encrypted watermark is embedded into the modified difference image. The modified difference image is scanned. Once a pixel with the difference value of -1 or 1 is encountered, we check the encrypted watermark to be embedded. If the bit to be embedded is 1, we move the difference value of -1 to -2 by subtracting one from the odd-line pixel or 1 to 2 by adding one to the odd-line pixel. If the bit to be embedded is 0, we skip the pixel of the difference image until a pixel with the difference value -1 or 1 is encountered. In this case, there is no change in the histogram. Therefore, the watermarked old-line field $I_w(i,2j+1)$ is obtained as follows: If $S(m,n) = 1$ and $\widetilde{D}(i,j) = 1$ or -1,

$$I_w(i,2j+1) = \begin{cases} \widetilde{I}(i,2j+1)+1 & \text{if } \widetilde{D}(i,j) = 1 \\ \widetilde{I}(i,2j+1)-1 & \text{if } \widetilde{D}(i,j) = -1 \end{cases} \tag{8}$$

and in all other cases,

$$I_w(i,2j+1) = \widetilde{I}(i,2j+1) \tag{9}$$

and the watermarked even-line fields $I_w(i,2j)$ is given by

$$I_w(i,2j) = I(i,2j) \tag{10}$$

The embedding capacity of this scheme equals to the number of pixels with the difference values of -1 and 1 in the difference image. A large number of pixel values of the difference image have a tendency to be distributed around 0. Using this property of the difference image, we can embed a large amount of data as compared to the original image itself.

Assume that there is no pixel with overflow and underflow in the original image. In the worst case, all pixels of the odd-line field will be added or subtracted by 1. The mean squared error of this case is 0.5. Hence, the PSNR of the watermarked image can be calculated as

$$\text{PSNR(dB)} = 10\log_{10}(255^2 \cdot 2) \approx 51.14 \tag{11}$$

In short, the lower bound of the PSNR of the watermarked image is about 51.14 dB. This result is much higher than other lossless data hiding techniques [2], [6], [7].

2.2. Integrity Verification

Fig. 2 depicts the integrity verification procedure of the proposed scheme. In this process, we verify that the image has not been tampered with and if the image is authentic, we reverse the watermarked image back to the original image.

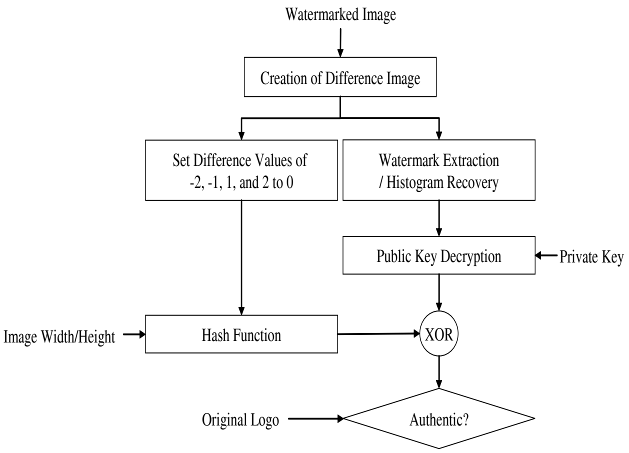

Watermarked Image
Creation of Difference Image
Set Difference Values of -2, -1, 1, and 2 to 0
Watermark Extraction / Histogram Recovery
Public Key Decryption
Private Key
Image Width/Height
Hash Function
XOR
Original Logo
Authentic?

Fig. 2. Integrity verification

For the received watermarked image $I_e(i,j)$ of size $M_e \times N_e$ pixels, we calculate the difference image $D_e(i,j)$ of size $M_e \times \frac{N_e}{2}$ pixels. The whole difference image is scanned. The encrypted watermark $S_e(m,n)$ is extracted as follows:

$$S_e(m,n) = \begin{cases} 0 & \text{if } D_e(i,j) = -1 \text{ or } 1 \\ 1 & \text{if } D_e(i,j) = -2 \text{ or } 2 \end{cases} \tag{12}$$

We use a public key decryption algorithm to decrypt $S_e(m,n)$ with the private key K_s that corresponds to the public key K_p used in the watermark embedding procedure. That is, we calculate

$$W_e(m,n) = Z_D(K_s, S_e(m,n)) \tag{13}$$

where $Z_D(\cdot)$ is the decryption function of the public key system.

We reverse the watermarked image back to the original image by shifting some pixel values in the difference image. The whole difference image is scanned once again. The recovered odd-line field $I_r(i,2j+1)$ can be expressed as

$$I_r(i,2j+1) = \begin{cases} I_e(i,2j+1) - 1 & \text{if } D_e(i,j) \geq 2 \\ I_e(i,2j+1) + 1 & \text{if } D_e(i,j) \leq -2 \\ I_e(i,2j+1) & \text{otherwise} \end{cases} \tag{14}$$

Since we manipulate pixel values of only the odd-line field in the watermark embedding process, the recovered even-line field $I_r(i,2j)$ is obtained by

$$I_r(i,2j) = I_e(i,2j) \tag{15}$$

Hence, the corresponding difference image $D_r(i,j)$ is calculated as

$$D_r(i,j) = I_r(i,2j+1) - I_r(i,2j) \tag{16}$$

Let

$$\begin{aligned} \mathbf{D_r} &= \{D_r(i,j) : 0 \leq i \leq M-1; 0 \leq j \leq \tfrac{N}{2} - 1\} \\ \tilde{\mathbf{D}}_\mathbf{r} &= \{\tilde{D}_r(i,j) : 0 \leq i \leq M-1; 0 \leq j \leq \tfrac{N}{2} - 1\} \end{aligned} \tag{17}$$

Then, we compute the hash $\hat{A}(k)$ of length 128 as

$$H(M_e, N_e, \tilde{\mathbf{D}}_\mathbf{r}) = \{\tilde{A}_e(k) : 0 \leq k \leq 127\} \tag{18}$$

where $\tilde{\mathbf{D}}_\mathbf{r}$ equals the corresponding element in $\mathbf{D_r}$ except difference values of -2, -1, 1 and 2 are set to zero.

In order to extract the embedded binary logo image of the image, we combine the hash $A_e(k)$ with the extracted watermark $W_e(m,n)$ of length C using the bit-wise XOR operation. In the case that $C > 128$, we form $\tilde{A}_e(k)$ by periodically replicating $A_e(k)$ to the length C. Then, the embedded binary logo image $\tilde{B}_e(m,n)$ of the length C is computed as

$$\tilde{B}_e(m,n) = \tilde{A}_e(k) \oplus W_e(m,n) \tag{19}$$

In the case that $C > P \times Q$, $\tilde{B}_e(m,n)$ is the replication version of $B_e(m,n)$.

Finally, we form $\tilde{B}(m,n)$ by periodically replicating $B(m,n)$ to the length C and verify the integrity of the image by comparing $\tilde{B}_e(m,n)$ with $\tilde{B}(m,n)$. If the image is authentic, we can restore the original image without any distortion.

$$D_r(i,j) = D(i,j) \text{ and } I_r(i,j) = I(i,j) \tag{20}$$

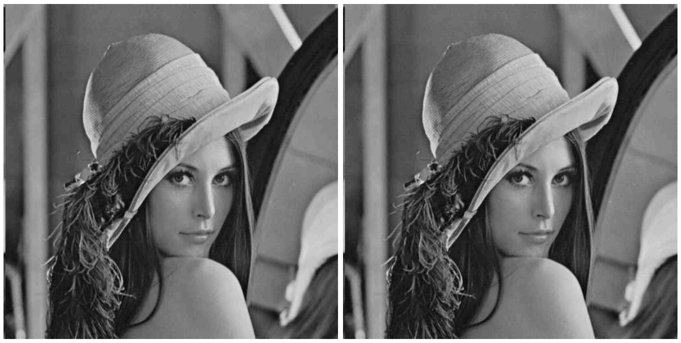
3. EXPERIMENTAL RESULTS AND ANALYSIS

In order to evaluate the performance of the proposed authentication scheme, we perform computer simulations on several test images of size 512 × 512 pixels. Fig. 3 shows a watermark which is a binary logo image of size 128 × 26 pixels, equivalent to a binary sequence of 3,328 bits.

Fig. 3. Binary logo image of 128 × 26 pixels

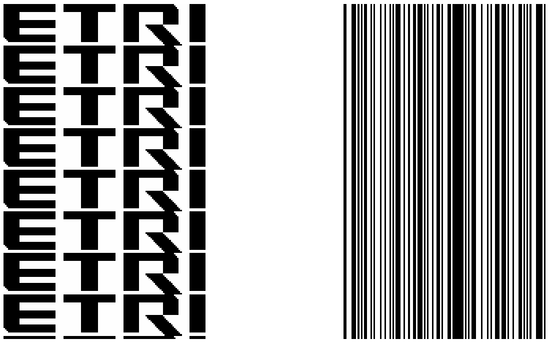
Fig. 4 shows the watermarked Lena image with the authentication watermark added by the proposed scheme. It is seen from the figure that since we modify pixel values slightly to embed the authentication watermark, there is no visible degradation due to embedding in the watermarked image.

Table 1 summarizes the experimental results. This table shows that the PSNR values of all watermarked images are above 51.14 dB, as we theoretically proved in Section 2.1. The capacity ranges from 8,150 bits to 28,094 bits for



(a) Original Lena image (b) Watermarked Lena image

Fig. 4. Watermarked image



(a) Expanded logo image (b) Wrong key used

Fig. 5. Extracted logo images on Lena

$512 \times 512 \times 8$ test grayscale images. This result shows that the proposed scheme offers adequate capacity to address most applications. It is also seen from Table 1 that an image like Baboon, which contains significant texture, has considerably lower capacity than simple images such as Airplane(F-16).

Table 1. PSNR and capacity of the proposed scheme

Test images	PSNR (dB)	Capacity (bits)
Lena	52.21	26,990
Airplane (F-16)	52.27	28,094
Baboon	51.41	8,150
Boat	51.65	14,503
House	52.12	23,126
Peppers	51.45	17,143

In our experiments, we have considered several attacks of pixel value changes in the watermarked image. if the watermarked image is not changed, and if the correct key is used in the watermark extraction process, the appropriate binary logo image is extracted, as shown in Fig. 5(a). In this case, we can reverse the watermarked image back to the original image. If the image is not watermarked or if the pixel values of the watermarked image are changed, the extracted binary logo image appears like random noise. If the inappropriate key is used in the extraction process, the embedded binary logo image is periodically deformed with period one line, as shown in Fig. 5(b). This is because of the properties of the MD5, that is, any string of data is hashed into a bit array of length 128.

4. CONCLUSIONS

In this paper, a new watermarking-based image authentication technique is presented. The technique provides low distortion, high embedding capacity, and lossless data hiding. Experimental results show that the proposed authentication scheme can detect if the key is incorrect, if the image is not watermarked, or if the image is changed in its pixel values. That is, we can verify the integrity of the image. Finally, we note that many authentication schemes based on watermarking have the ability to identify regions of the image that have been corrupted. We are currently studying this localization capability for our proposed scheme.

5. ACKNOWLEDGMENT

This work was supported in part by IITA and MIC through RBRC.

6. REFERENCES

[1] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, 2001.

[2] J. Fridrich, M. Goldjan, and R. Du, “Invertible authentication,” *Proc. SPIE, Security and Watermarking of Multimedia Contents*, pp. 197–208, Jan. 2001.

[3] S. Lee, Y. Suh, and Y. Ho, “Reversible image authentication based on watermarking,” *Proc. ICME 2006*, will be presented in July 2006.

[4] R. Rivest, “The MD5 message digest algorithm,” Internet RFC 1321, 1992.

[5] R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining ditigal signatures and public-key cryptosystems,” *Commun. ACM*, pp. 120–126, Feb. 1978.

[6] Z. Ni, Y. Shi, N. Ansari, and W. Su, “Reversible data hiding,” *Proc. ISCAS 2003*, vol. 2, pp. 912–915, 2003.

[7] J. Tian, “High capacity reversible data embedding and content authentication,” *Proc. ICASSP*, pp. III-517–520, 2003.