Proceeding of the 2008 International Joint Conference in Engineering
IJSE2008, August 4-5, Jakarta, Indonesia

A Public-Key Watermarking Method for Still Images Based on Random Permutation

Rinaldi Munir*, Bambang Riyanto*, Sarwono Sutikno*, Wiseto P. Agung †

* School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia,
Jl. Ganesha 10, Bandung 40132
Tel.+6222 25078135, fax. +6222 2539172, email: rinaldi@if.itb.ac.id
† PT. Telekomunikasi Indonesia, Jl. Gegerkalong Bandung

Abstract– **We present a public-key watermarking method for still images based on random permutation. This method uses secret watermark as private key and public watermark as public key. The secret watermark is obtained by permuting the public watermark. The embedded watermark is a linear combination of the secret and public watermark The watermark is embedded into mid-frequency components of the DCT block for balancing between image fidelity and robustness. The detection step is implemented by correlation test between the public watermark and the received image. Experiments show that the watermarking method was proved to be robust againts various attacks using common image processings (cropping, JPEG compression, resizing, rotation, sharpening, and noising).**

Keywords– image, public-key watermarking, permutation, correlation, DCT, robust

I. INTRODUCTION

Digital data multimedia (e.g images) are easily copied, stored and transmitted over computer network, internet and ohter media. However these properties lead to problem enforcing copyright protection, for example illegal copy or unauthorized use. Digital watermarking has been proposed for the copyright protection of the digital multimedia data [1]. Basic requirements of a digital watermarking scheme are 1) imperceptibility: a watermark is inserted into digital images so that it is imperceptible to a person, 2) robustness: the watermark must be robust to typical signal processing operations such as JPEG compression, cropping, resizing, noising, rotation, and so on, 3) security: the watermark should also only be accessible by authorized parties [2].

Many digital watermarking methods for still images have been proposed [1-3]. The particular problem with the state-of-the-art watermarking methods is that the majority of these schemes are symmetric. The *symmetric watermarking* is similar to symmetric cryptography: the same key is used for watermark embedding and detection. The symmetric watermarking scheme has a security problem. In many watermarking methods, the secret key represents the watermark itself or specifies the embedding location of the watermark. Because the watermarking algorithm is published, once attacker knows the secret key, the watermark can detected, and addition, be easily estimated and removed from the multimedia data compeletely without making any degradation and thereby defeat the goal of copyright protection.

A solution to solve the problem is the *asymmetric watermarkin*g scheme, in which different key(s) are used for watermark embedding and detection. Asymmetric watermarking is called *public-key watermarking* if key used for detecting watermark is available publicly, so the key is called *public key*. Suppose that the host signal $\mathbf{x} = (x_0, x_1, \ldots, x_{m-1})$ serves as the carrier for the watermark $\mathbf{w} = (w_0, w_1, \ldots, w_{n-1})$. The host signal may be of pixels of the original image or transform coefficients extracted from the original image. Fig. 1 depicts a general public-key watermarking scheme. The watermark **w** is embedded into the host signal dependent on a private key. The watermaked signal is **y** that can be expressed as $\mathbf{y} = \mathbf{x} + \mathbf{w}$. The detection step is done by using a public key and a binary output decision generated (the received signal contains the watermark or not).

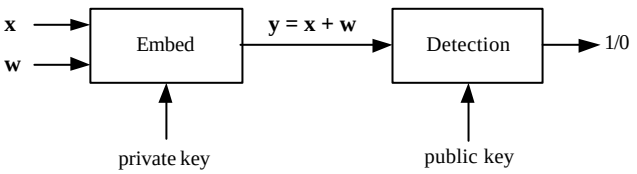


Fig. 1. General public-key watermarking scheme.

Only the copyright holder has his *private key* to embed a watermark. Anybody who knows the public key could detect the watermark. Therefore, public-key watermarking can provide public detection, but the secret key cannot be

deduced from the public key. Also, knowing the public key does not enable an attacker to remove the watermark [3]. Some public-key watermarking methods have been proposed in recent years. Review of several existing methods can be found in [4].

Generally, in public-key watermarking scheme, the secret key is a secret (or privat) watemark embedded into host media and the public key is a public watermark. To enable detection, the public watermark should have a correlation with the secret watermark. The detection step is implemented by correlation test between public watermark and multimedia data received [6]. Comparing the detection value with a predefined threshold, a decision should be made to decide the presence of the embedded watermark.

There exist numerous methods to generate the private watermark that are different but have a fixe d correlation with the public watermark. One of them is by using permutation. In this paper we present a public-key watermarking method based on a random permutation. We use a permutation table to obtain the private watermark from the public watermark. We embed the private watermark into still images in the DCT domain..Next, we test robustness of the proposed technique againts various attacks using common image processings (JPEG compression, cropping, resizing, etc).

II. WATERMARKING IN DCT DOMAIN

Current image watermarking methods can be grouped into spatial domain methods and transform domain methods. In spatial domain, we embed the watermark by directly modifyng the pixel values of the original image. In transform domain, a transformation is first applied to the original image and then embedding the watermark into transform coefficient. There are three main transform methods generally used, i.e Fourier transform (DFT), discrete cosine transform (DCT), and wavelet transform (DWT). Embedding the watermark into the transform-domain can increase the robustness, when the watermarked image are tested after having been subjected to common image processings. In this paper we transform the original image using DCT method.

The DCT can be applied to transform the whole image or image blocks (8 × 8 pixel). By referring to JPEG compression, watermarking that operates on 8×8-DCT blocks yields better robustness than that on the whole image [12].

The DCT allows an image to be divided into different frequency subbands: low frequency, middle frequency, and high frequency (see Fig. 2 for 8×8-DCT blocks). Embedding

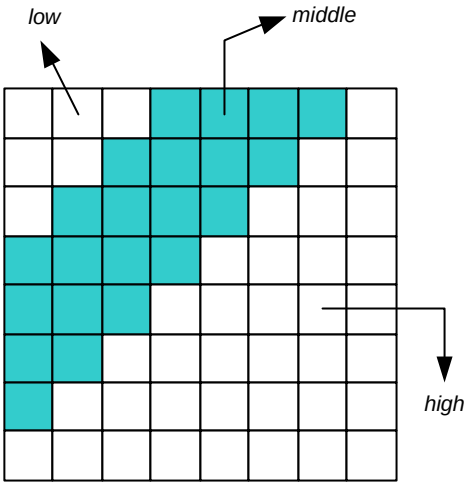


Fig. 2. Definition of DCT regions

the watermark into the low-frequency subbands coefficient can degrade the image quality, whereas high frequency components are easily discarded after low pass filterring or JPEG compression. Therefore, for balancing between image fidelity and robustness, most watermarking techniques embed the watermark into the middle-frequency subbands coefficients.

III. THE PROPOSED METHOD

We present an asymmetric watermarking technique for still images based on random permutation. There are three stages in this technique: the watermark generating, the watermark embedding, and the watermark detection. Each stage will be explained in next sub-sections.

III.1 Generation of Watermark

The watermark size is about 25% of the original image size. Therefore, if the image size is $N_1 \times N_2$ pixel, then watermark size is about $N_1 N_2 / 4$. The watermark is a real sequence that chosen according to a normal distribution with *mean* = 0 and *variance* = 1 (notation: N(0, 1)).

Firstly, generate a public watermark $\mathbf{w_p}$ whose length is n according to normal distribution N(0, 1):

$$\mathbf{w_p} = (w_p(1), w_p(2), ..., w_p(n))$$

Next, we construct a secret watermark $\mathbf{w_s}$ by permuting $\mathbf{w_p}$ according to a permutation table S:

$$\mathbf{w_s} = (w_s(1), w_s(2), ..., w_s(n))$$
$$= (w_p(S(1)), w_p(S(2)), ..., w_p(S(n)))$$

The public watermark is published, but the permutaiton table *S* and an initial value of the logistic map must be keept secret.

III.2 Embedding of Watermark

The watermark embedding scheme is combination of [5] and [12] schemes. The embedded watermark is a linear combination of the secret and public watermark. This watermark is computed by formula [5]:

$$\mathbf{w_e} = (1 - \alpha)\mathbf{w_s} + \alpha\mathbf{w_p} \qquad (1)$$

where the term α is a weighted factor to control the public detection theshold, and $0 < \alpha < 1$.

The original image is divided into small blocks with 8 × 8 pixel. Next, apply the DCT for every block, then the DCT coefficients of the block -except DC value- are scanned by zigzag order to extract mid-frequency components. Suppose the selected components is represented by **x**, then the embedded watermark $\mathbf{w_e}$ is inserted into **x** by formula [12]:

$$x_w(i) = x(i) + \gamma |x(i)| w_e(i) \qquad (2)$$

where γ is a watermark strength constant that is adjusted to make the watermark imperceptible. Finally, using IDCT (inverse of the DCT), we get the watermarked image.

III.3 Watermark Detection

The proposed detection technique is not require the original image and the secret watermark. Detector requires only the public watermark that has a correlation with the secret watermark. Given a received image and a public watermark, only two cases are possible: the image contains the watermark or the image does not contains the watermark.

Watermark detection is done in the following steps. Firstly, the received image is divided into small block with with 8 × 8 pixel. Next, apply the DCT for every block and then the DCT coefficients of the block, except DC value, are scanned by zigzag order to extract mid-frequency components. Suppose the selected components is represented by **x***, then the correlation between **x*** and the public watermark $\mathbf{w_p}$ is computed by formula:

$$c = \frac{1}{N}\sum_{i=i}^{N} x^*(i) \cdot w_p(i) \qquad (3)$$

This correlation is compared to a threshold *T*: if $|c| > T$, we say a watermark signal exists; otherwise, a watermark signal does not exist. The threshold *T* is derived empirically by examining the correlation of random sequences.

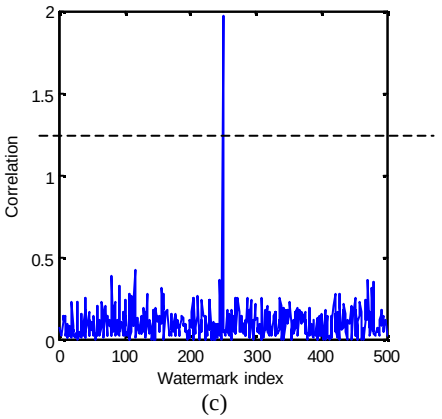
IV. EXPERIMENT AND RESULTS

We program the watermarking algorithm above by using MATLAB 7. The test image is a 256 × 256 gray image 'Barbara. The public watermark is a 128 × 128 real matrix that has a normal distribution with *mean* = 0 and *variance* = 1. We use $\alpha = 0.4$ and $\gamma = 0.6$. We use function `randperm` to produce a random permutation table.

Figure 3a shows the original image and Figure 3b shows the watermarked image (*PSNR* = 38.3261). Next, we derive the detection threshold empirically. Figure 3c shows the detection threshold of 500 random public watermarks studied, and only one public watermark, which has a correlation with the secret watermark, has a significantly higher correlation output than the others. The threshold *T* is set to be 1.25 in this graph (dashed line). In case no attack done, the detector results $c = 1.9701$. This value is greater than the *T*, it means the received image contains the watermark.

(a)

(b)



(c)

Fig. 3. (a) Original image. (b) Watermarked image. (c) Detection threshold experimently. *T* is set to be 1.2.

If the received image does not contain the watermark (in this experiment we use an unwatermarked 'Barbara' image as input to detector), we get $c = 0.0427$ and there is not a significantly higher correlation output than the others (Fig. 4). We conclude that the image does not contain the watermark.

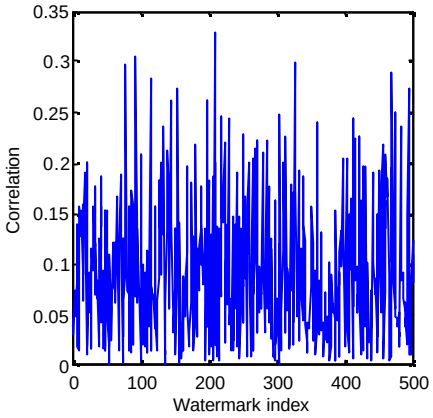
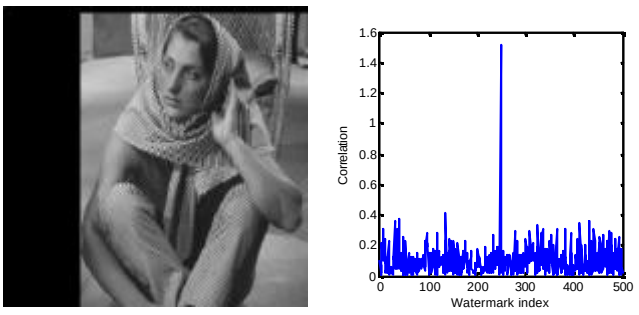


Fig. 4. There is not a significanty higher correlation output than the others. The test image does not contain the watermark.

We have tested robustness of the proposed method againts various attacks using common image processings (JPEG compression, cropping, resizing, etc). We use *Jasc Paint Shop version 6.01* as image processing software. For every attack, we set different thresholds, depend on experiment to derive the threshold empiracally. The experiment and results are explained as follows.

IV.1 Experiment 1: Cropping

Image cropping will remove some watermark information. In our simulation, we cut unimportant part from the watermarked image, the missing part of the image is replaced with black pixels (see Figure 5a). In fact, we can always corectly detect the watermark because the correlation value (c = 1.5076) is still greater than T. In this case, we set T = 1.0 from experiment (see Figure 5b).

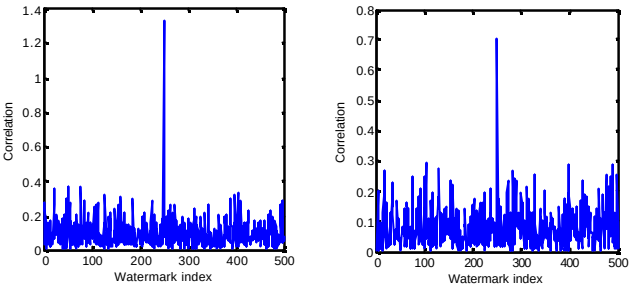


(a) (b)

Fig. 5 . (a) Cropping image. (b) Detector response. The watermark still can be detected (we set T = 1.0).

IV.2 Experiment 2: JPEG Compression

We tested the robustness against JPEG compression with various extreme compression qualities: 20% and 10%. In this experiment we use MATLAB to get JPEG file. To detect the watermark, the JPEG files is returned to bitmap versions. The watermark still can be detected for the various extreme compression quality (the detector results a significantly higher correlation than random watermarks, see Fig. 6 for details).

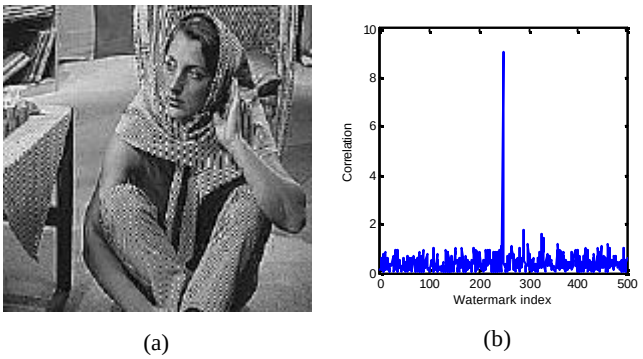


(a) (b)

Fig 6. JPEG compression with various extreme compression qualites. (a) 20%. (b) 10%. The watermark still can be detected

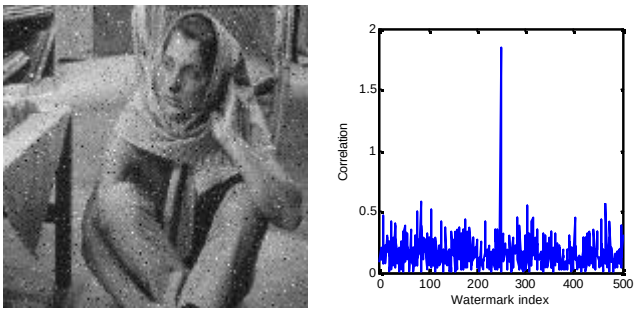
IV.3 Experiment 3: Sharpening and Adding Noise

The watermarked image is sharpened some times until their edges bok sharper than the original version. We still detect the presence of the watermark (see Fig. 7, in this case we set T = 6.0 and c = 9.0739). We also add some noises like salt and peppers of 10%. The results show that the watermark can be detected (see Fig. 8, in this case we set T = 1.25 and c = 1.8355).



(a) (b)

Fig. 7. (a) Image sharpening. (b) Detector response.

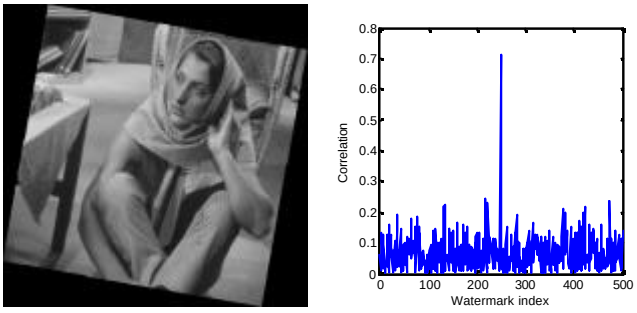


(a) (b)

Fig. 8. (a) Adding noise. (b) Detector response

IV.4 Experiment 4: Rotation

The watermarked image is rotated by as many as 10° (Figure 9), then to detect the watermark, the rotated image is returned to original position. It turns out that we still can detect the watermark. (c = 0.7123, we set T = 0.5).

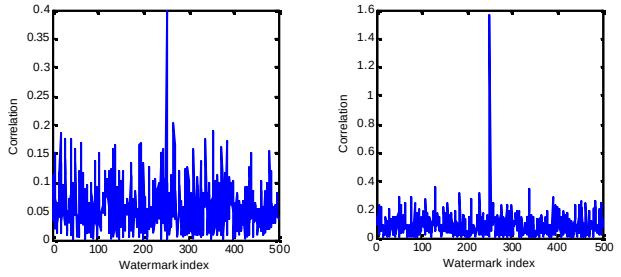


(a) (b)

Fig. 9. (a) Image rotation (10°). (b) Detector response.

IV.5 Experiment 5: Resizing

The image watermarked is resized until 50% of the original size. To detect the watermark, the smaller image is returned to original size. We found c = 0.3996 and the watermark still can be detected (see Fig. 10a). For resizing up to 200% of the original image, the watermark still can be detected (see Fig. 10b).



(a) (b)

Fig. 10. Image resizing (a) 50% and (b) 200%.

V. CONCLUSION

In this paper a public-key watermarking method based on random permutation has been proposed. The private watermark is produced by permuting the public watermark. The detection process is implemented by correlation test between public watermark and features of the image received. Simulation have confirmed that this technique is robust againts non-malicious attacks (cropping, JPEG compression, resizing, rotation, sharpening, and noising).

REFERENCES

{1] Ingemar J. Cox, dkk, "*Secure Spread Spectrum Watermarking for Multimedia*", IEEE Trans. On Image Processing, Vol. 6, No. 12, Dec 1997, pp.1673-1687.

[2] I. Wiseto P. Agung, *Watermarking and Content Protection for Digital Images and Video*, thesis of PhD in University of Surrey, 2002.

[3] Mauro Barni, Franco Bartolini, *Watermarking Systems Engineering*, Marcel Dekker Publishing, 2004.

[4] Joachim J. Eggers, Jonathan K. Su, and Bernd Girod, *Asymmetric Watermarking Schemes*, GMD Jahrestagung, Proceddings, Springer-Verlag, 2000.

[5] G.F Gui, L.G Jiang, C He, *General Construction of Asymmetric Watermarking Based on Permutation*, Proc. IEEE Int. Workshop VLSI Design & Video Tech., May 28, 2005.

[6] T.T. Kim, T. Kim, dan H. Choi, *Correlation-Based Asymmetric Watermarking Detector*, Int. ITCC, 2003.

[7] H. Choi, K. Lee, dan T. Kim, *Transformed-Key Asymmetric Watermarking System*, IEEE Signal Processing Letters, Vol. 11. No. 2, February 2004.

[8] Zhao Dawei, Chen Guanrong, Liu Wenbo, "*A Chaos-Based Robust Wavelet-Dmain Watermarking Algorithm*", Jurnal Chaos Solitons and Fractals 22 (2004) 47-54.

[9] www.yahoo.com, *Chaos Theory: A Brief Introduction*, diakses pada bulan November 2005

[10] James Lampton, *Chaos Cryptography: Protecting Data Using Chaos*, Mississippi School for Mathematics and Science.

[11] Hongxia Wang, Chen He, and Ke Ding, "*Public Watermarking Based on Chaotic Map*", IEICE Trans. Fundamentals, Vol. E87-A, No. August 2004.

[12] Sangoh Jeong and Kihyun Hong, *Dual Detection of A Watermark Embedded in the DCT Domain*, EE368A Project Report, 2001.

[13] Yong-Gang Fu, Rui Min Shen, Li Ping Shen, *A Novel Asymmetric Watermarking Scheme*, Proc. of the 3rd Int. Confererence on Machine Learning and Cybernetics, 2004.

[14] Guo Fu Gui, Ling Ge Jiang, and Chen He, *A New Asymmetric Watermarking Scheme for Copyright Protection*. IECE Trans. Fundamentals, Vol. E89-A, No. 2 February 2006.