Derivation of Barni Algorithm into Its Asymmetric Watermarking Technique Using Statistical Approach

Rinaldi Munir[1], Bambang Riyanto T[2], Sarwono Sutikno[3], Wiseto P. Agung[4]

[1, 2, 3]School of Electrical Engineering and Informatics,
Bandung Institute of Technology, Indonesia,
[4]PT. Telekomunikasi Indonesia, Jl. Gegerkalong Bandung
[1]rinaldi@stei.itb.ac.id,
[2]briyanto@lskk.ee.itb.ac.id,
[3]ssarwono@ieee.org
[4]wiseto@telkom.co.id

Abstract: This paper presents an asymmetric watermarking technique derived from Barni Algorithm, a symmetric watermarking technique, using statistical approach. This asymmetric version uses secret watermark as private key and public watermark as public key. The public watermark has a normal distribution and the private watermark is a linear combination of the public watermark and encrypted version of a secret sequence. The detection process is implemented by correlation test between the public watermark and the received image. Experiments show that the asymmetric technique was proved as robust as its symmtric version againts some typical image processings.
Keywords: asymmetric watermarking, Barni Algorithm, derivation, correlation.

1. Introduction

Digital watermarking has been used widely as a tool for protecting copyright of digital multimedia data (e.g images) [1, 2]. Many digital watermarking techniques for still images have been proposed [1-3]. The particular problem with the state-of-the-art watermarking techniques is that the majority of these schemes are symmetric: watermark embedding and detection use the same key. The symmetric watermarking scheme has a security problem. Because the watermarking algorithm is published, once attacker knows the secret key, the watermark not only can be detected, but it can 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 watermark*ing scheme, in which different key(s) are used for watermark embedding and detection. An asymmetric watermarking system uses the *private key* to embed a watermark and another key – it is called the *public key* – to verify the watermark. Anybody who knows the public key could detect the watermark, but the private key cannot be deduced from the public key. Also, knowing the public key does not enable an attacker to remove the watermark [3].

Suppose that the host signal $X = (x_0, x_1, \ldots, x_{m-1})$ serves as the carrier for the watermark $W = (w_0, w_1, \ldots, w_{n-1})$. The host signal may be of pixels of the original image or transform coefficients extracted from the original image. Fig. 1 depicts a general asymmetric 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 $Y = X + W$. The detection step is done by using a public key and a binary output decision generated (the received signal contains the watermark or not). Review of several existing asymmetric watermarking techniques can be found in [4].

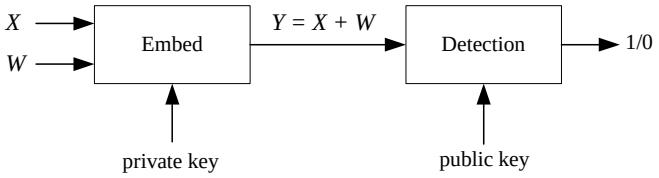


Fig. 1. General asymmetric watermarking scheme

We need intensive effort and time if we design a new asymmetric watermarking technique. Thus we think to derive a symmetric watermarking technique into its asymmetric version. Key of this transformation is based on process of generating the private key and the public key. In this paper, we choose an existing symmetric watermarking technique achieving very good results in robustess. We choose Barni Algorithm [5] because the algorithm achieve very good results in imperceptibility and robustness. We will compare performance between the symmetric watermarking and its asymmetric version.

2. 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 before embedding watermark. Then, the transform coefficients are modified to embed the watermark and finally the inverse transform is applied to obtain the watermarked image. Since the watermark embedded in the transform domain is irregularly distributed over the image after the inverse transform, the method make it more difficult for an attacker to read or modify the watermark. Futhermore, 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.

There are three main transform methods generally used, i.e Fourier transform (DFT), discrete cosine transform (DCT), and wavelet transform (DWT). In this paper we use 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 [6].

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

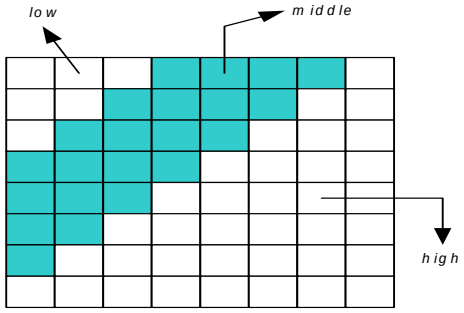


Fig. 2. Definition of DCT regions

3. Barni Algorithm

Barni *et al* [5] propose image watermarking system which the original un-watermarked image is not needed in the detection stage. The watermark consists of a pseudo random sequence of M real number, $W = \{w_1, w_2, ..., w_M\}$, that has a normal distribution with *mean* = 0 and *variance* = 1.

The original image I is transformed by DCT, then the DCT coefficients are scanned by zigzag order, such as in the JPEG compression algorithm, and the mid-frequency coefficients ar extracted by selecting the first $L+M$ coefficients. Suppose the selected components is represented by $X = \{x_1, x_2, ..., x_M\}$ then the watermark W is inserted into V by formula:

$$x_w(i) = x(i) + \alpha \left| x(i) \right| w(i) \qquad (1)$$

where α 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.

Watermark detection is done in the following steps. Given a possibly corrupted image I^*, the DCT is applied to I^*, then the DCT coefficients are scanned by zigzag ordering to extract mid-frequency components. Suppose the selected components is represented by $X^* = \{x_1^*, x_2^*, ..., x_M^*\}$, then the correlation between X^* and the watermark W is computed by formula:

$$c = \frac{1}{M} \sum_{i=i}^{M} x^*(i) \cdot w(i) \qquad (2)$$

This correlation is compared to a threshold T: if $\left| c \right| > T$, we say a watermark signal exists; otherwise, a watermark signal does not exist.

4. Asymmetric Technique

We derive an asymmetric technique from Barni Algorithm. Key of this transformation is based on process of generating the private key and the public key. The public key should have a correlation with the private key. The private watermark is embedded into the image. User can perform an asymmetric detection using a correlation test between the public watermark and the received image.

In Barni Algorithm, the secret key is the watermark itself where it has normal distribution. In asymmetric version of Barni Algorithm, the private key and the public key is referred as the private watermark and the public watermark. We want the two watermarks to have normal distribution.

There exist numerous methods to generate the private watermark that are different but have a fixed correlation with the public watermark and both watermarks have normal distribution. One of them is by using statistics approach. In statistics, if we add two or more random variables as a linear combination where each of them has normal distribution, then the result has normal distribution too. Let X be a sequence with mean μ_1 and variance σ_1^2 and Y be sequence that independent from X with mean μ_2 and variance σ_2^2. A combination linear of X and Y is defines as

$$Z = aX + bY \qquad (3)$$

where a and b is parameters. Sequence Z has the mean

$$\mu_3 = a\mu_1 + b\mu_2 \qquad (4)$$

and variance

$$\sigma_3^2 = a^2\sigma_1^2 + b^2\sigma_2^2 \quad (5)$$

In generating the watermarks we have to ensure that the combination linear is secure. It means that the private watermark cannot be deduced from the public watermark. Also, knowing the public watermark does not enable an user to remove the embedded watermark from the watermarked image. This characteristic is realized by adding the public watermark with a secret sequence. Security of this asymmetric version depend on the secret sequence. Let W_p be the public watermark and R be the secret sequence, the private watermark can be obtained by adding W_p and R as

$$W_s = \beta W_p + (1 - \beta) R \quad (6)$$

where β is a parameter in [0, 1] to control the compromise between the two sequences. In order to make the sequence R is more secure, we encrypt R by a random permutation before adding with W_p. Thus, eq. (6) can be written as

$$W_s = \beta W_p + (1 - \beta) \tilde{R} \quad (7)$$

where $\tilde{R}$ is encrypted version of R. Fig. 3 shows process of generating the public and the private watermark.

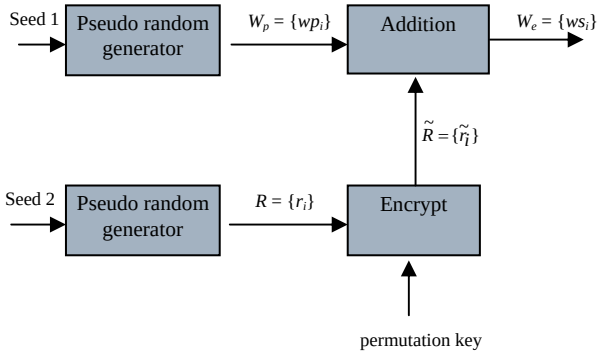


Fig. 3. Generating of the public and the private watermark

In asymmetric version, we use two watermarks, the first is a private watermark that embedded into the host image and the second is a public watermark for detection phase. The both watermarks is generated by the procedure explained in Section 2. The private watermark is embedded into the image according to eq. (4) by replacing W with W_S:

$$x_w(i) = x(i) + \alpha \,|\, x(i) \,|\, w_s(i) \quad (8)$$

In the detector side, using the public watermark, W_p, the following correlatin is computed:

$$c = \frac{1}{M} \sum_{i=i}^{M} x^*(i) \cdot w_p(i) \quad (9)$$

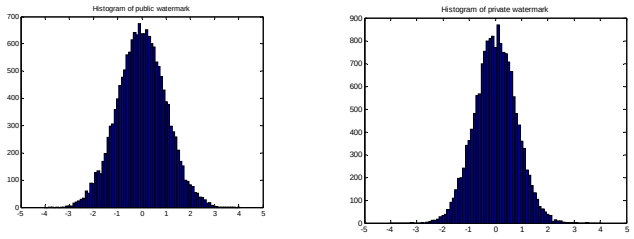
After we set the threshold T, the watermark detection is finished by the comparison between c and the threshold.

5. Security Analysis

If an attacker want to remove the watermark from the watermaked image, he (or she) must find $\tilde{R}$ in order to get W_s according to equation (7). Once W_s is calculated, the original image can be extracted by manipulation of equation (8). The attacker knows W_p, α and β but he (or she) does'nt know $\tilde{R}$. Because $\tilde{R}$ is encrypted version of R, the attacker hard to find it. Let the attacker knows R, next he (or she) need know a random permutation used to encrypt R. Because cardinality of R is n, the attacker must try $C(n, n) = n!$ permutation to find the right permutation. Remember that n is large enough, it is about 25% of original image size, so that finding the right permutation needs $O(n!)$ computation. For n = 10000 as example, there are 10000! computation! We conclude it is impossible for attacker to deduce the private watermark from these public information.

6. Experiment And Results

We apply our method to image watermarking by using MATLAB as programming tool. The test image is a 512 × 512 color image 'Fajar'. The public watermark is a 128 × 128 real matrix that has a normal distribution with *mean* = 0 and *variance* = 1. The embedding strength α is equal to 0.2 and parameter β is equal to 0.75. Histogram of the public watermark and the private watermark is shown in Fig. 4. From Fig. 4(b) we observe that shape of distribution graphics of the private watermark is like a bell as common standard normal distributions.



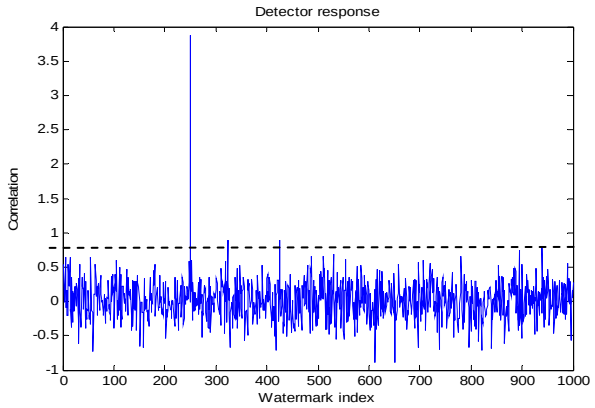
(a) (b)

Fig.4. Histogram of the public and private watermark

Before embedding the private watermark, the original image is transformed from RGB to YcbCr. The watermark is embedded to luminance component (Y) only, and the final result is retransformed from YcbCr to RGB. Figure 5(a) shows the original image and Figure 5(b) shows the watermarked image (*PSNR* = 38.9637). Next, we derive the detection threshold empirically. Figure 5(c) shows the detection threshold of 1000 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 = 3.8806. This value is greater than the T, it means that the received image contains the watermark. As comparison, if we detect by using the private watermark, it results correlation c = 3.2345. It means performance of the asymmetric technique is above its symmetric version.

If the received image does not contain the watermark (in this experiment we use an unwatermarked 'Fajar' image as input to detector), we get c = 0.1302 and there is no a significantly higher correlation output than the others (Fig. 6). We conclude that the image does not contain the watermark.

(a) (b)



(c)

Fig. 5. (a) Original image. (b) Watermarked image.
(c) Detection threshold experimently. *T* is set to be 0.75.

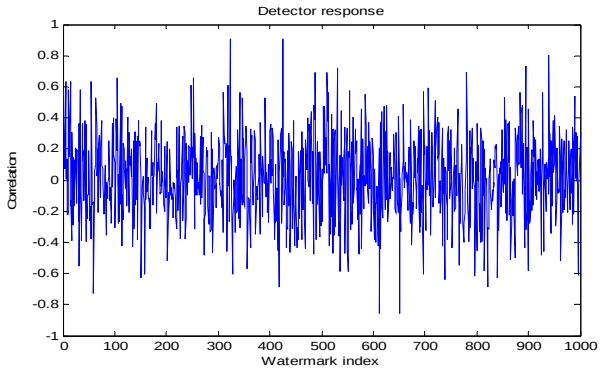


Fig.6. There is no a significanty higher correlation output than the others. The test image does not contain the watermark.

We have tested robustness of our 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 the experiment to derive the threshold empirically. The experiments and results are explained as follows.

6. 1 Experiment 1: JPEG Compression

We tested the robustness against JPEG compression with extreme compression qualitie. For compression quality 5%, the watermark can be detected well (c = 1.4720). The detector shows a significantly higher correlation than random watermarks, see Fig. 7 for details.

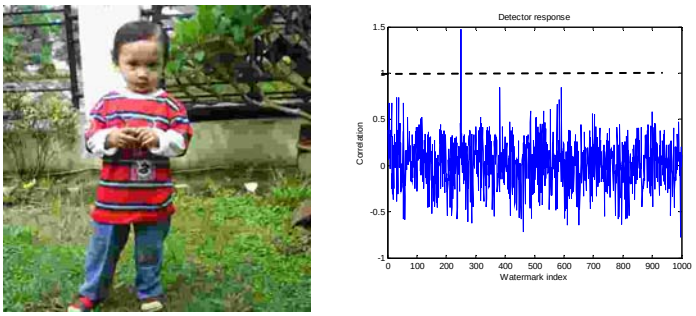
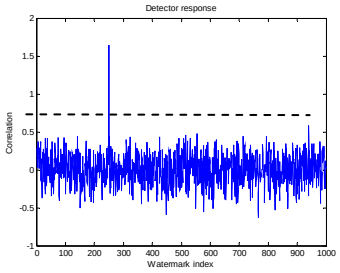


Fig 7. JPEG compression with compression quality 5%. The watermark can be detected

6.2 Experiment 2: Image Cropping

Image cropping will remove some watermark information. In our simulation, we cut unimportant part from the watermarked image (about 50%), the missing part of the image is replaced with black pixels (see Figure 8(a)). In fact, we can always corectly detect the watermark because the correlation value (c = 1.6448) is still greater than T. In this case, we set T = 0.75 from the experiment (see Figure 8(b)).

(a)



(b)

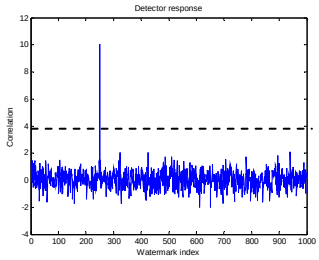
Fig. 8. (a) Image cropping. (b) Detector response.

The watermark still can be detected (we set T = 0.75).

6.3 Experiment 3: Sharpening and Adding Noise

The watermarked image is sharpened several times (high-pass filtering operation) until their edges look sharper than the original version. We still detect the presence of the watermark (see Fig. 9, in this case we set T = 4.0 and c = 10.0735). We also add some noises like salt and peppers of 50%. The results show that the watermark can be detected (see Fig. 10, in this case we set T = 1.5 and c = 3.3007).

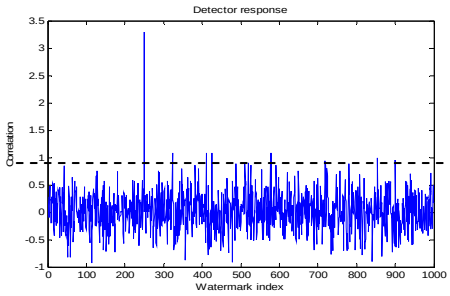
(a)



(b)

Fig. 9. (a) Image sharpening. (b) Detector response.

(a)

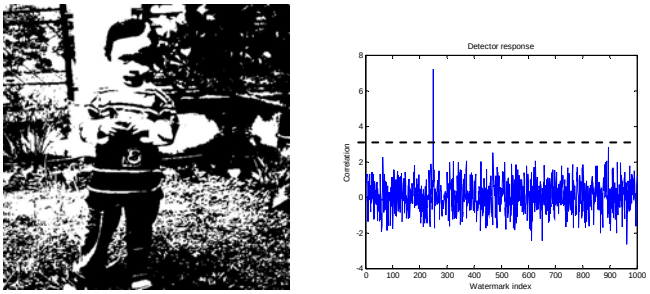


(b)

Fig. 10. (a) Adding noise. (b) Detector response

6.4 Experiment 4: Dithering

We convert the watermarked image to a binary image by dithering operation. It means plenty of gray-level information lost. It is shown in Fig. 11 that the watermark still can be detected. The response to the right watermark is largest among the response to all the watermarks (c = 7.2139).

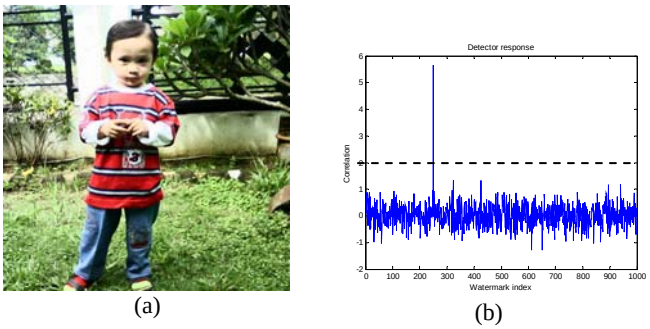


(a) (b)

Fig. 11. (a) Dithering. (b) Detector response

6.5 Experiment 5: Histogram Equalization

The watermarked image is adjusted so that distribution of gray-level is uniform by using histogram equalization operation (a typical low-pass filtering operation). Experiment shows that the watermark can be detected where c = 5.6694 and T = 2.0 (see Fig. 12).

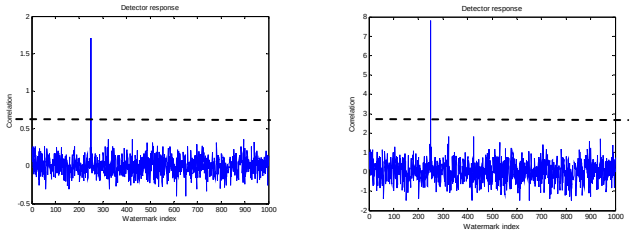


(a) (b)

Fig. 12. (a) Histogram equalization. (b) Detector response

6.6 Experiment 6: Resizing

The watermarked image is resized until 50% of the original size. To detect the watermark, the smaller image must be returned to original size (else the watermark can not be detected). We found that c = 1.7095 (we set T = 0.6) and this experiment shows that the watermark still can be detected (see Fig. 13(a)). For resizing up to 200% of the original image, the watermark still can be detected well (we found that c = 7.7992 and we set T = 3.0) (see Fig. 13(b)).

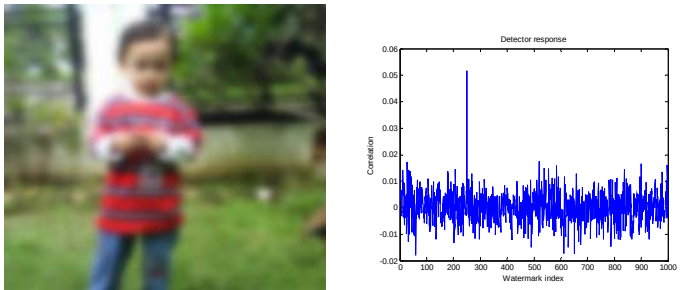


(a) (b)

Fig. 13. Image resizing (a) 50% and (b) 200%.

6.7 Experiment 7: Blurring

The watermarked image is blurred with Gaussian blurring. Experiment shows that the watermark still can be detected (Fig. 14).



(a) (b)

Fig. 14. Image blurring

7. Conclusion

In this paper an asymmetric watermarking technique for still images derived from its symmetric version has been proposed. This technique uses two watermarks; the first watermark is a public watermark used to public detection, and the second watermark is a private watermark that has a correlation to the public watermark. The private watermark is a linear combination of the public watermark and a encrypted version of a secret sequence. Security of this asymmetric technique is based on the difficulty of finding the secret sequence where it needs O(n!) computation. Simulation have confirmed that this asymmetric technique is as robust as its symmetric version.

References

[1] Ingemar J. Cox, et al, "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] Mauro Barni, F. Bartolini, V. Cappellini, A.Piva, "A DCT-Domain System for Robust Image Watermarking", *Signal Processing* 66, pp 357-372, 1998.

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

[7] Peter Meerwald, Digital Image Watermarking in The Wavelet Transform Domain, *diploma thesis in Salzburg Univrsity*, 2001.