An Asymmetric Watermarking Method in the DCT Domain Based on RC4-Permutation and Chaotic Map

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

[1,2,3]Bandung Institute of Technology, Jl. Ganesha 10,Bandung 40132 , Indonesia
[4]PT. Telekomunikasi Indonesia, Jl. Gegerkalong, Bandung, Indonesia

Abstract. This paper presents an asymmetric watermarking method in the DCT domain for still images based on permutation and chaos. This method uses secret watermark as private key and public watermark as public key. The public watermark has a normal distribution with mean = 0 and variance = 1. The secret watermark is obtained by permutating the public watermark according to combination of a part of RC4 algorithm and a logistic map. The watermark is embedded into mid-frequency components of the DCT block for better robustness. The detection process 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 some typical image processings (cropping, JPEG compression, resizing, rotation, sharpening, and noising).

Keywords: *image, asymmetric, watermarking, RC4, chaos, correlation, DCT, robust.*

1 Introduction

Recently, with the emergence of computer network and internet, lots of digital multimedia data are easily copied, stored and transmitted over the world, leading to illegal copy or unauthorized use. Digital watermarking has been used widely as a tool for protecting copyright of digital multimedia data (e.g images) [1]. Basic requirements of a digital watermarking scheme are imperceptibility, robustness, and security [2]. A watermark is inserted into digital images so that it is imperceptible to a person. The watermark must be robust to typical signal processing operations such as JPEG compression, cropping, resizing, noising, rotation, and so on. The watermark should also only be accessible by authorized parties.

Many digital watermarking methods for still images have been proposed [1-3]. The particular problem with the state-of-the-art watermarking method 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 an 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 an *asymmetric watermark*ing scheme, in which different key(s) are used for watermark embedding and detection. An asymmetric watermarking system uses the secret key to embed a watermark and another key to verify the watermark. It is not necessary to detector to know the seceret key. An asymmetric watermarking is called *public-key watermarking* if key used for detecting watermark is available publicly, so the key is called *public key*. Only the copyright holder has his *private key* to embed a watermark. Anybody who knows the public key could detect the watermark. Therefore, an asymmetric 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]. Asymmetric watermarking methods have been proposed in recent years. Review of several existing methods can be found in [4].

Generally, in asymmetric watermarking scheme, the secret key is a secret (or privat) watermark 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 fixed correlation with the public watermark. One of them is by using permutation. In this paper we present an asymmetric watermarking method based on permutation. We use a combination of a part of RC4 cipher and a chaotic map as permutation method. A chaotic map is used to produce a pseudo-random signal. In recent years, chaos have been used for digital watermarking to increase security [8], due to its sensitivity to initial condition. This method has high robustness, and it is secure to the malicious attack (the attack which its objective is removing the watermark from the watermarked image.

In comparison to existing related methods [13-14], in this paper we propose a combination of RC4-permutation and chaotic map for asymmetric watermarking method. In this method the watermark is embedded into the transform-domain (DCT domain). This paper is organized as follows. In Section 2 we will present chaotic function, in Section 3 we present RC4 algorithm, and

in Section 4 we present watermarking in DCT domain. Next, in Section 5 we present the asymmetric watermarking method proposed, in Section 6 we report the experimental results, and finally some conclusions in Section 8.

2 Chaos Function

One of the characteristic of chaotic systems is a sensitivity to initial conditions; i.e two relatively close initial value will diverge as the system evolves. As a result of this sensitivity, the behavior of systems appears to be random, even though the system is deterministic; i.e it is well defined and contains no random parameters. Hence, a chaotic system can be used as a pseudo-random generator. It means a large number of non-periodic, noise-like yet deterministic and reproducible sequences can be generated [11].

We consider a 1-D discrete chaotic map $\mathbf{F} : U \rightarrow U$, $U \subset \mathbf{R}$, which provides sequence of real number:

$$x_{k+1} = \mathbf{F}(x_k, \lambda) \ , \ x_k \in U, \lambda \in \mathbf{R} \qquad (1)$$

where $n = 0, 1, 2, \ldots$ denotes map iterations and λ is a parameter that controls the dynamic behavior of the chaotic map. In our scheme, an initial value of chaotic map behave as the key of the watermarking system.

One of the simplest chaotic maps is a *logistic map* [8], which is a recurrence relation that describes population growth over time, desribed by

$$x_{k+1} = \mu x_k (1 - x_k) \qquad (2)$$

where $0 \leq \mu \leq 4$. When we iterate equation (2) from an initial value (x_0), we get a chaotic sequence. Elements of this sequence are between 0 and 1.

3 Permutation Based on RC4

RC4 (also known as ARC4 or ARCFOUR), one of cryptography algorithm (cipher), is the most popular stream cipher. RC4 generates a pseudorandom stream of bits (a keystream) which is combined with the plaintext using XOR (encryption) or with the ciphertext (decryption). This algorithm uses array $S[0..255]$ which is initialized with 0, 1, …, 255. Before encryption/decryption process, elements of S are permutated based on external key U. Length of U is

variable, but if length of *U* less than 256 characters, we must do *padding* so that its length is equal to 256 characters. Algorithm for permutating elements of S is as follows:

```
j ← 0
for i ← 0 to 255 do
    j ← (j + S[i] + U[i]) mod 256
    swap(S[i], S[j])
end
```

This algorithm is modified to permutate the public watermark. Suppose that length of the watermark is *N*, then elements of *S* are initialized with 0, 1, …, *N*. Key *U*, which its length is *N*, is an integer array whose elements are generated by a logistic map (each chaos value is multiplied by *N* and then rounded to the nearest integer). Next, elements of *S* are permutated with the algorithm except 255 is replaced by $N - 1$. The values in *S* are used to permutate the public watermark.

4 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 modifiyng 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 is chosen because it is simple and mainly for its assonance with JPEG coding standard.

The Two dimensional *DCT* of an *M* by *N* matrix is defined as follows [2]:

$$C(p,q) = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) \cos \frac{\pi 2(m+1)p}{2M} \cos \frac{\pi (2n+1)q}{2N} \qquad (3)$$

and the inverse DCT (or IDCT) is given by

$$I(m,n) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \alpha_p \alpha_q C(p,q) \cos \frac{\pi 2(m+1)p}{2M} \cos \frac{\pi (2n+1)q}{2N} \qquad (4)$$

where

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}} & , p = 0 \\ \sqrt{\frac{2}{M}} & , 1 \le p \le M-1 \end{cases} \qquad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}} & , q = 0 \\ \sqrt{\frac{2}{N}} & , 1 \le q \le N-1 \end{cases}$$

The values $C(p, q)$ are called the DCT coefficients of image I.

The DCT can be applied to transform the whole image or image blocks (8×8 pixel). By referring to JPEG compression that operates on 8×8-DCT blocks, a watermarking that operates on the same block size 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 Fifure 1). Embedding 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 the image fidelity and robustness, most watermarking techniques embed the watermark into the middle-frequency subbands coefficients.

5 The Proposed Method

We present an asymmetric watermarking technique for still images based on permutation using RC4 algorithm and chaotic map. 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.

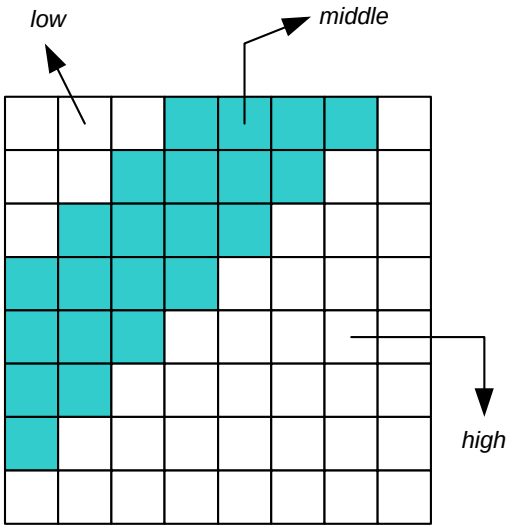


Figure 1 Definition of DCT regions.

5.1 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 sequence of real numbers 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 permutating $\mathbf{w_p}$ according to permutation table S that is taken from RC4 algorithm (see Ssection 3):

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

The public watermark is published, but the permutation table S and the initial value of the logistic map must be kept secret.

5.2 Embedding of Watermark

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

$$\mathbf{w_e} = (1 - \alpha)\mathbf{w_s} + \alpha\mathbf{w_p} \quad (5)$$

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

The original image I is divided into small blocks of 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 **f**, then the embedded watermark $\mathbf{w_e}$ is inserted into **f** by formula [12]:

$$f_w(i) = f(i) + \gamma \left| f(i) \right| w_e(i) \quad (6)$$

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.

5.3 Watermark Detection

The proposed detection technique does 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 contain the watermark.

Watermark detection is done in the following steps. Firstly, the received image is divided into small block of 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 ordering to extract mid-frequency components. Suppose the selected components is represented by **f***, then the correlation between **f*** and the public watermark $\mathbf{w_p}$ is computed by formula:

$$c = \frac{1}{N}\sum_{i=i}^{N} f^*(i) \cdot w_p(i) \quad (7)$$

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.

5.4 Security Analysis

If an attacker want to remove the watermark from the watermaked image, he (or she) does it by manipulation of equation (4) to get $f(i)$ as follows:

$$f(i) = \frac{f_w(i)}{1 \pm \gamma w_e(i)} \tag{8}$$

The attacker knows $f_w(i)$, $\mathbf{w_p}$, and γ, but he (or she) must have $\mathbf{w_s}$ in order to compute w_e by using equation (5). Because $\mathbf{w_s}$ is get by a secret permutation S, whereas S also depends on a secret chaotic sequence, then the attacker can not calculate (8). If the attacker try to generate the chaotic sequence, he (or she) must try all possibility of initial values of the logistic map. Since the logistic map is sensitive to initial value, the attacker will fail to discover the chaotic sequence; in other words, the attacker it is impossible for attacker to deduce the private watermark from the public information.

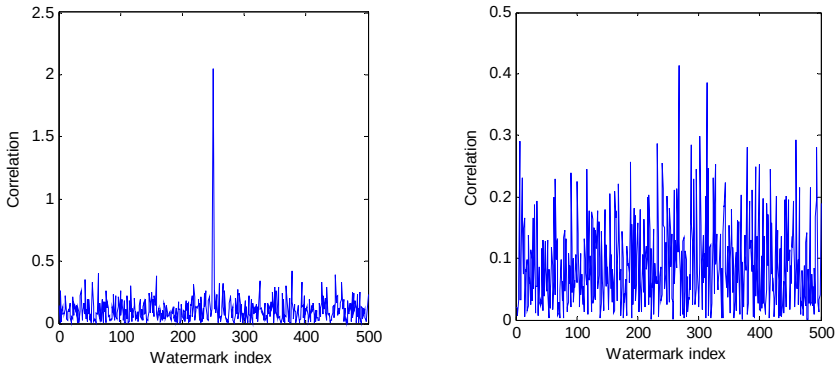
6 Experiments and Results

We program the watermarking algorithm using MATLAB 7. The test image is a 256 × 256 gray image 'peppers'. The public watermark is a 128 × 128 real matrix that has a normal distribution with *mean* = 0 and *variance* = 1. A chaotic sequence is generated by logistic map with the initial value $x_0 = 0.45$. We use $\alpha = 0.4$ and $\gamma = 0.6$. And 500 random watermarks were generated for evaluating detection method.

Figure 2(a) shows the original image and Figure 2(b) shows the watermarked image (*PSNR* = 37.8044). Next, we derive the detection threshold empirically. Figure 3(a) shows the detection threshold of 500 random public watermarks studied. Only one public watermark, which has a correlation with the secret watermark, has a significantly higher correlation output than the others. In case no attack done, the detector results $c = 2.0507$.

(a) (b)

Figure 2 (a) Original image. (b) Watermarked image.

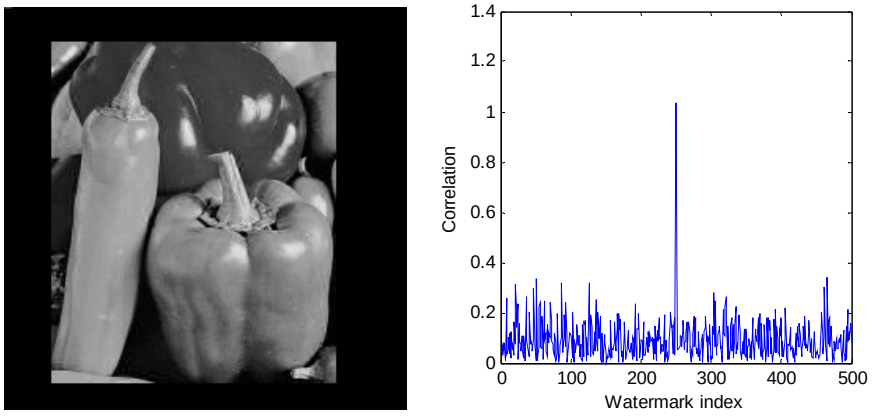


(a) (b)

Figure 3 Detection threshold experimently; (a) case watermark exists, and (b) case no watermark exist.

If the received image does not contain the watermark (in this experiment we use an unwatermarked 'peppers' image as input to detector), we get $c = 0.1299$ and there is not a significantly higher correlation output than the others (Fig. 3(b)). We conclude that the image does not contain the watermark.

We have tested robustness of the proposed technique 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.



(a) (b)

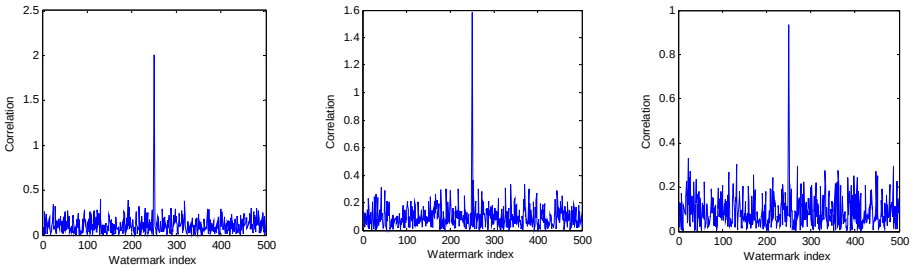
Figure 4 (a) Cropping image. (b) Detector response. The watermark still can be detected.

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 4a). When the watermarked image is cropped, the correlation value is decreased (c = 1.0339) but this value is still significantly higher than the others (see Figure 4b). And we can see that the watermarking is robust against image cropping.

Experiment 2: JPEG Compression

We tested the robustness against JPEG compression with various compression qualities: 80%, 20%, and 10% (extreme quality). In this experiment we use MATLAB to get JPEG file. To detect the watermark, the JPEG files is returned to bitmap versions. The correlation values are decreased (Fig. 5). However, the correlation values are still above random and we conclude the watermark is robust against JPEG compression.

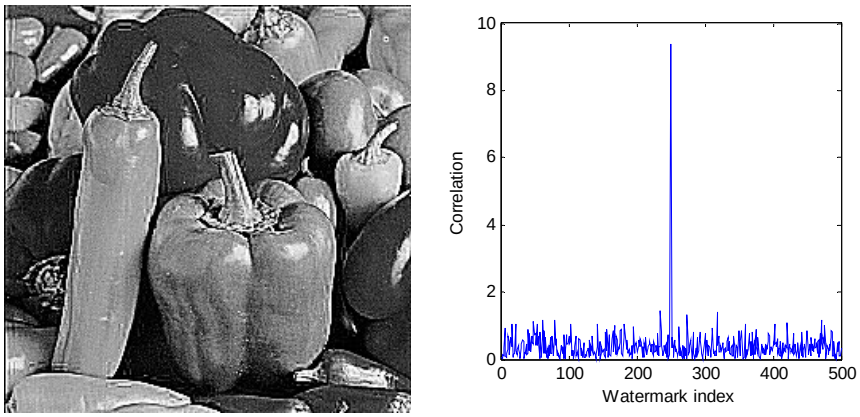


(a) Quality 80% (b) Quality 20% (c) Quality 10%

Figure 5 JPEG compression with various qualites. The watermark still can be detected.

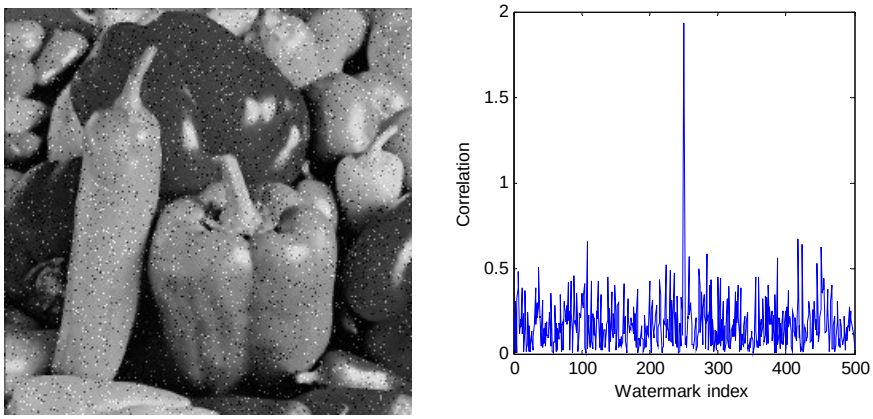
Experiment 3: Sharpening and Adding Noise

The watermarked image is sharpened until their edges look sharper than the original version. The detector shows that the correlation value is increased (c = 9.3838) and it is significantly above random (see Fig. 6). We also add some noises like salt and peppers of 10%. The results show that the watermark can be detected (see Fig. 7, c = 1.9302). We conclude from these experiments that the watermarking is robust againts image sharpening and adding noise.



(a) (b)

Figure 6 (a) Image sharpening. (b) Detector response.



(a) (b)

Figure 7 (a) Adding noise. (b) Detector response.

Experiment 4: Rotation

When a image is rotated, the DCT coefficient alignment is changed. It cause that the watermark can not be detactable. In order to the watermark can be detected, the rotated image must be returned to original position. In this experiment the watermarked image is rotated by as many as 10° (Fig 8), then to detect the watermark, the rotated image is returned to original position. It turns out that the correlation value is still above random (c = 1.09) and we conclude that the watermark can be detected.

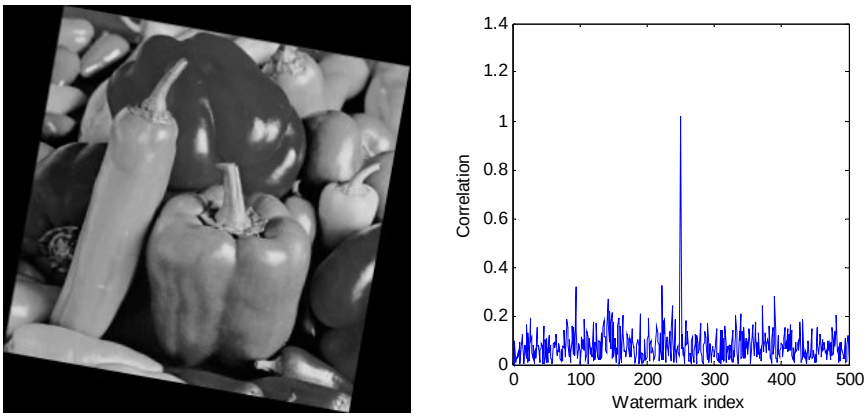
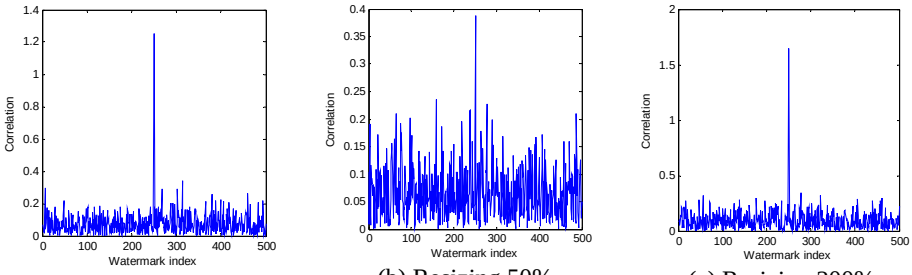


Figure 8 Image rotation with angle 10°. The watermark still can be detected.

Experiment 5: Resizing

The image watermarked is resized 75%, 50%, and 200% of the original size respectively. To detect the watermark, the new image is returned to original size. We can see that the correlation values are still significantly higher than the others. We conclude that the watermarking is robust against image resizing. .



(a) Resizing 75% (b) Resizing 50% (c) Resizing 200%

Figure 9 Image resizing 75%. The watermark still can be detected.

7 Conclusion

In this paper an asymmetric watermarking technique for still images based on permutation-RC4 and chaotic map has been proposed. This technique applies the DCT to image blocks with 8×8 pixel and embed watermark into mid-frequency components. The private watermark is produced by permutating the public watermark according to a part of RC4 algorithm and logistic map. 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.