Synchronous Counterfeiting Attacks on self-embedding Watermarking Schemes

Hongjie He †, Jiashu Zhang †, and Hongxia Wang ††

† Sichuan Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu 610031,China
†† School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031,China

Summary
This paper points out secure holes of the self-embedding watermarking schemes. To further illustrate their insecurity, this paper proposes a novel synchronous counterfeiting (SC) attack on self-embedding watermarking schemes. Some experimental results show that the proposed SC attack can not only forge watermark in any un-watermarked image and change the size of watermarked image randomly; but it also generate the excellent quality of the counterfeit watermarked image by cropping and/or pasting watermarked images with image block as a unit.
Key words:
Self-embedding watermarking, Counterfeiting attacks, Vector quantization attack, the offset.

1. Introduction

Along with the digitization of image information and the appearances of powerful image editing programs, it is very easy even for an amateur to create "perfect" forgeries. It is also possible to cut out portions of several images and combine them together while leaving barely detectable traces [1]. Therefore, it is difficult to establish the genuine of digital image by perceptual inspection. Digital watermarking technology is the most popular technique for establishing image integrity and authenticity.

According to the integrity criteria, the watermark-based image authentication techniques can be divided into hard authentication and soft authentication [2]. Hard authentication rejects any modifications to image content; while soft authentication is capable of distinguishing malicious changes from innocent image operations. Different types of watermarks have been proposed in the literature designed for different applications. In some applications, it is often desirable to design an authentication system which may provide more information such as tamper locations [3, 4], severity [4, 5], and approximate recovery when a watermarked image is determined not to be authentic. These functions have been fulfilled by self-embedding watermarking schemes [1, 6-7].

In 1999, Fridrich [1] proposed a DCT-based self-embedding watermarking scheme firstly, in which the image was divided into 8×8 blocks that were DCT transformed, quantized, and carefully encoded into the LSBs of other distant 8×8 blocks according to the fixed offset p. In 2004, Zhang etc. [7] put forward a new DCT-based self-embedding scheme, in which the number of coefficients and their bit lengths were carefully chosen according to the statistical property of these coefficients and the method for choosing the offset was analyzed to improve security. The self-embedding watermarking algorithm embeds watermark into the LSB of another block with an offset that increases the dependence among the image blocks, therefore, this algorithm can not only resist VQ attack effectively [9, 10], but it also recovers portions of images that have been cropped or replaced or severely modified [7]. However, the key space of offset in self-embedding watermarking algorithms is too small, and the watermark can be detached from the content of watermarked images. These two weaknesses make it possible to find out the corresponding watermark of image block in watermarked image. If the attacker tampers watermarked image block content (7MSBs) while adjusts the corresponding watermark (LSB) according to the offset synchronously, the tampered image is possible to be as authentic. That has brought serious security holes to the self-embedding watermarking algorithms.

In this paper, we firstly analyze the security of the current self-embedding schemes, and then propose a synchronous counterfeiting attack on them. We show that self-embedding schemes are potentially vulnerable to attacks whereby counterfeit watermarks can be inserted into images without the commission of the authenticator. Specifically, given one or more watermarked images using a same insertion key and un-watermarked image $\mathbf{Z}$, it is possible for an attacker to construct a watermarked image $\mathbf{Z}'$ with the similar visual quality to image $\mathbf{Z}$. Moreover, the excellent quality of counterfeit watermarked image could be obtained by cropping and/or collage several watermarked images.

Manuscript revised December 22, 2005.

The rest of this paper is organized as follows. The security of self-embedding watermarking algorithms [1, 7] is discussed in section 2. In section 3, we propose the novel synchronous counterfeiting attacks on self-embedding watermarking algorithms, and demonstrate successful the attacks on current self-embedding watermarking schemes. Finally, section 4 concludes this paper.

2. Security Analyses of Current Self-embedding Algorithms

In this section, we firstly introduce the basic idea of self-embedding algorithms presented in [1, 7]. And then the security hole of them has been demonstrated by theoretical analyses under the condition of known offset. Lastly, the possibility of getting offset is discussed by the watermarked image.

2.1 Description of current self-embedding watermarking algorithms

Self-embedding watermarking algorithm is one kind of the recovery watermarking techniques. The main content of an image as a watermark is embedded into itself in this kind of algorithms. It could not only detect and localize the altered areas, but it also recovers the missing information. In current self-embedding algorithms, watermark is often generated by using JPEG compression standard for reference.

2.1.1 Embedding algorithm

Step1 The image generated by setting LSB of each pixel in the original image $X_{m\times n}$ to zero is called $\widetilde{X}$, and then $\widetilde{X}$ is partitioned into non-overlapping blocks, noted as $\widetilde{x}_i$, i=1,2,…,N, where N represents the number of blocks;

Step2 Each block $\widetilde{x}_i$ is transformed by DCT, quantized, and carefully encoded to generate the binary sequence m_i. Then the m_i is encrypted to generate watermark block w_i to be embedded, depicted as:

$$m_i = f(\widetilde{x}_i) \tag{1}$$

$$w_i = E_{k_e}(m_i) = E_{k_e}(f(\widetilde{x}_i)) \tag{2}$$

Where, *f(.)* represents the process of DCT transform and quantization encoding, *E(.)* is the encryption function, k_e is a insertion key.

Step3 Watermark block w_i is inserted into the LSBs of the block $\widetilde{x}_{i+p_i}$ according to the offset $\vec{p} = \{p_i, i = 1,2,..,N\}$ to produce watermarked image block, defined as:

$$y_{i+p_i} = \widetilde{x}_{i+p_i} + w_i \tag{3}$$

2.1.2 Tamper detection and recovery

Step1 By the same method as Step1 in embedding algorithm, divide the tested image $\boldsymbol{Y}^*$ into non-overlapping blocks y_i^*, i=1,2,…,N ;

Step2 The binary sequence of recovery from LSBs of the i^{th} block is obtained by extraction key k_d, i.e.:

$$ml_i^* = D_{k_d}(LSB(y_i^*)) \tag{4}$$

Where, *LSB(.)* means the least significant bit plane of image block.

Step3 The binary sequence of generating by the content (7MSBs) of image block y_i^* is obtained in a similar way to the sender:

$$m_i^* = f(\widetilde{y}_i^*) \tag{5}$$

Step4 Tamper detection and localization is performed by comparing binary sequence m_i^* and $ml_{i+p_i}^*$ according to corresponding offset $\vec{p}$:

·If $m_i^* = ml_{i+p_i}^*$, then block y_i^* and $y_{i+p_i}^*$ are both not tampered;

·If $m_i^* \neq ml_{i+p_i}^*$, then either y_i^* or $y_{i+p_i}^*$ is altered by the situation y_i^* and $y_{i+p_i}^*$ for reference [7].

Step5 If the detection results show block y_i^* is tampered, then the watermark in un-tampered image block $y_{i+p_i}^*$ by decoded, inverse quantization and IDCT transform, could be used to reconstruct the tampered block y_i^* approximately [7].

2.2 Security Holes of Self-embedding Algorithms

It can be known from above description, the security of self-embedding algorithms depend on the encryption process *E(.)* of generating watermark, the corresponding watermark of the image block could not be obtained without encryption secret key k_e, which could resist the known counterfeiting attacks. At the same time, the watermark is embedded into LSB of another image block with an offset $\vec{p}$, which enhances the relativity among image blocks. Consequently, this kind of schemes could resist VQ attack [8] and collage attack [9].

However, the following characteristics of self-embedding algorithm produce secure holes:

(1) The watermark is embedded into LSB of every pixel in watermarked image, so it is easy to detach the watermark from the image content;

(2) In order to have superior recovery quality, the watermark w_i should contain the information about

image block as much as possible, i.e., w_i depends only on the content of image block y_i;

(3) Deduce whether the content of image block y_i is tampered or not, it's only related to the LSB of image block $y^*_{i+p_i}$.

(4) Under the condition of known watermarked image, it is possible to obtain the offset key by adopting some methods such as the verification device attack (details discuss in next sub-section).

In order to discuss the security holes brought by these characteristics, watermarked blocks are denoted by $y_i = \tilde{y}_i + \ddot{y}_i$ (i=1,2,…,N), where $\tilde{y}_i$ represents image content (7MSBs) of y_i, $\ddot{y}_i$ means the LSB plans of y_i. The block structure of watermarked image is shown in figure 1. Clearly, the corresponding watermark w_i of y_i, which is used to validate the integrity and authenticity of $\tilde{y}_i$, is equal to $\ddot{y}_{i+p_i}$, i.e. $w_i = \ddot{y}_{i+p_i}$.

$\tilde{y}_1$	$\tilde{y}_2$	…	$\tilde{y}_i$	…	$\tilde{y}_{i+p_i}$	…	$\tilde{y}_{N-1}$	$\tilde{y}_N$
$\ddot{y}_1$	$\ddot{y}_2$		$\ddot{y}_i$		$\ddot{y}_{i+p_i}$		$\ddot{y}_{N-1}$	$\ddot{y}_N$

$i + p_i$

Fig. 1 Block structure of the watermarked image

Accordingly, for arbitrary watermarked block y_i (i=1,2,…,N), the following equation comes into existence:

$$\begin{cases} f(\tilde{y}_i) = m_i \\ D_{k_d}(w_i) = D_{k_d}(\ddot{y}_{i+p}) = m_i \end{cases}, \quad \forall i \in [1, N] \tag{6}$$

Therefore

$$f(\tilde{y}_i) = D_{k_d}(\ddot{y}_{i+p_i}) \tag{7}$$

That is to say, the tested image could be considered as authentic if only every block in it is satisfied with formula (7). Consequently, the modified image could not be detected if the content and corresponding watermark of image block is synchronously replaced by the watermarked block. The details of tamper are shown in the following:

For the sake of simplicity and without loss of generality, given y_i and y_j($i, j \in [1, N] \& i \neq j$) are two different watermarked blocks in watermarked image $\boldsymbol{Y}$, $\vec{p}$ is the offset. The altered image, noted as Y', is obtained by $\mathrm{Re}\,lpace(y_i, y_j, \vec{p})$, which includes two following operations:

$$\begin{cases} \tilde{y}_i = \tilde{y}_j \\ \ddot{y}_{i+p_i} = \ddot{y}_{j+p_i} \end{cases} \tag{8}$$

We call $\mathrm{Re}\,lpace(y_i, y_j, \vec{p})$ as the synchronous replacement operation because of modifying image block content and corresponding watermark synchronously.

Proof:

Clearly, the alteration of $\ddot{y}_{i+p_i} = \ddot{y}_{j+p_i}$ is so slight that perceptual quality is preserved. However the manipulation $\tilde{y}_i = \tilde{y}_j$ has been changed the content of block y_i (7MSBs) which is the most significant part. That deduces the perceptual quality of Y' is not as same as $\boldsymbol{Y}$'s, i.e. $Y' \neq Y$. Now the proof will be given to prove the watermarking algorithm can not detect these modifications.

According to formula (1), the binary sequence generated by the i^{th} block in the altered image is:

$$m_i' = f(\tilde{y}_j) \tag{9}$$

The watermark embedding positions of blocks y_j and y_i are calculated as $i+p_i$ and $j+p_i$ by the offset $\vec{p}$, respectively. Using formula (8), the binary sequence recovered by the corresponding watermark of the i^{th} block in the altered image is:

$$ml'_{i+p_i} = D_{k_d}(\ddot{y}_{j+p_j}) \tag{10}$$

By combining with formula (7), (9) and (10), $m_i' = ml'_{i+p_i}$ is concluded

According to Step4 in detection algorithm, the detector will deduce that there is no tamper to blocks y_i and y_{i+p_i} in image Y', that is, the watermarking detection algorithm could not detect this kind of tamper.

The proof finishes.

The above property means that given a watermarked image block y_i and corresponding watermark w_i, the attacker can put this block in any position of watermarked image as long as the relative position of $\tilde{y}_i$ and w_i is unchanged according to the offset $\vec{p}$. Furthermore, the operation $\mathrm{Re}\,lpace(y_i, y_j, \vec{p})$ is independent of watermarked image and size of image. Therefore, during the operation $\mathrm{Re}\,place(y_i, y_j, \vec{p})$, the two different blocks y_i and y_j could be in the same or different watermarked images. The size of different watermarked images could be the same or not.

2.3 Possibility of Obtaining the Offset

According to the analyses in sub-section 2.2, if an attacker obtains a watermarked image and knows the offset, the altered image generated by the synchronous replacement operation could be considered as authentic. Therefore, the key space of offset should be big enough to guarantee the algorithm security. Unfortunately, the offset key space is too small and potentially obtained in current self-embedding algorithms [1, 7]. In this sub-section, from the key space standpoint, we will

1, 2, ……, *i*, …… …… *N-1*, *N*

discuss the possibility of getting the offset according to watermarked image.

In self-embedding algorithm, the offset $\vec{p}$ can be used to determine absolutely the embedding position of watermark, i.e., the embedding position $f(i)$ of the i^{th} watermark block is:

$$f(i) = i + p_i, \ i=1,2,\ldots,N \quad (11)$$

Obviously, *f(i)* should meet two requirements:

(1) The watermark of each image block must be embedded into LSB of some image block in the image, i.e. $f(i) \in [1, N]$, where $\forall i \in [1, N]$.

(2) The watermark from different blocks must not be embedded into the LSB of same block, i.e. $f(i) \neq f(j)$, where $\forall i, j \in [1, N] \& i \neq j$.

The following descriptions analyze the possibilities of getting offset in [1] and [7], respectively.

2.3.1 Self-embedding scheme by Fridrich

In [1], the offset $\vec{p}$ is a vector of length approximately 3/10 of the image size with a randomly chosen direction. In order to meet above two requirements, the offset direction in the same watermarked image should be the same. If the offset is considered as secret key, there are only two choices for the offset key. Obviously, the offset could be easily obtained according to watermarked image.

2.3.2 Self-embedding scheme by Zhang etc.

In [7], the watermark embedding position is:

$$f(i) = (k_0 + k_1 * i) \bmod N, \ i=1,2,\ldots,N \quad (12)$$

Where k_0 and k_1 are the offset keys, and k_1 should meet the requirement of $\boldsymbol{gcd}(k_1,N)=1$ [7], $\boldsymbol{gcd}(.)$ is the greatest common divisor.

The offset keys k_0 and k_1 are the coefficients of linear equation. If an attacker gets a watermarked image, it is possible to estimate the offset key by the method shown as follows:

(1) Using Verification Device Attack [2]

The attacker firstly modifies the watermark of the n_1^{th} block, and then the tampered watermarked image is submitted to the verification device. Here, the self-embedding watermarking schemes can detect the alteration and recover the "missing" information. In recovery results, there must be and only one block different from watermarked image. If the i_1^{th} block is different, it illuminates that the watermark of i_1^{th} block is embedded into the LSB of the n_1^{th} block. According to formula (12) it is easily made out: $n_1=(k_0+k_1*i_1) mod N$. Using the same method, another image block i_2 and the embedding position n_2 of corresponding watermark are achieved. It could get the following equation group:

$$\begin{cases} n_1 = (k_0 + k_1 * i_1) \bmod N \\ n_2 = (k_0 + k_1 * i_2) \bmod N \end{cases} \quad (13)$$

The offset keys k_0 and k_1 will be accurately calculated by resolving the equation group. This result once more shows linearity characteristic as secret key is the vulnerability to malicious attack.

(2) Using Exhaustive Attack

When the key space is not big enough, it is possible to obtain the key by exhaustive attack. The key space analyses of offset in [7] are shown as follows:

The selectable range of k_0 and k_1 is the integer in the interval within [0, N-1], and k_1 meets the requirement of $\boldsymbol{gcd}(k_1,N)=1$. Consequently, the offset key space is:

$$K = \phi(N) * N \quad (14)$$

Where, $\phi(.)$ is Euler function [10], its value equals to the number of integers which are prime with N and less than N.

Clearly, the number of image blocks N is not prime number. Given the unique prime decomposition of N is $N = p_1^{e_1} p_2^{e_2} \ldots p_k^{e_k}$, according to Euler theorem [10]:

$$\phi(N) = N(1-\frac{1}{p_1})(1-\frac{1}{p_2})\ldots(1-\frac{1}{p_k}) \quad (15)$$

According to formula (14) and (15), the offset key space has the close relation to the number of image blocks N. For example, for a gray scale image with size of 280×208, then N=5*7*2*13=910. According to formula (15) it could be gotten $\phi(N) \approx 0.316N$. Therefore, the offset key space is computed by formula (13) and is about $0.316*910^2 \approx 2.6*10^5$. Using the same calculating method, the offset key space of a image with size of 256×256 is about $219 \approx 5.2*10^5$. The key space of 10^5 is so small that it is easy to implement exhaustive attack with the computing speed of current computers, which results in successful estimation of the offset key. If the attacker obtains one or more watermarked images generated by same insertion key, the offset key could be achieved by certain method. Hence the synchronous replacement operation $\mathrm{Re}\,lpace(y_i, y_j, \vec{p})$ can actually be done quite successfully for some self-embedding watermarking techniques. The attacker could forge the watermark contained in given image by introducing the idea of vector quantization attack [8].

3. Synchronous Counterfeiting Attacks

To further illustrate the security holes of the existing self-embedding watermarking algorithms, herein, we propose the synchronous counterfeiting attacks on them.

3.1 Description of Synchronous Counterfeiting Attacks

Assume k_e is the insertion key, and k_p is the offset key,

some watermarked images $Y^n = \{y_1^n, y_2^n, ..., y_{N_n}^n\}$ (n=1,2,…,*num*) are generated by same key, where *num* is the number of watermarked images. The watermark embedding positions are denoted as $f(i,k_p,N_n)$, which relates to the image block position i, the offset key k_p and the number of image blocks N_n.

Given an un-watermarked image $Z = \{z_1, z_2, ..., z_N\}$, an attacker could construct a counterfeit watermarked image $Z' = \{z_1', z_2', ..., z_N'\}$ without the knowledge of the insertion key k_e, and Z' has the same visual quality as $\boldsymbol{Z}$, noted as $Z' \approx Z$. The process of synchronous counterfeiting attacks can be represented by the following generic procedure:

Input:

Watermarked image blocks $Y = \{y_1^1, ..., y_{N_1}^1, ..., y_1^n, ..., y_{N_n}^n, ...y_1^{num}, ..., y_{N_{num}}^{num}\}$;

Un-watermarked image $Z = \{z_1, z_2, ..., z_N\}$ in which we would like to forge watermark;

Output:

Counterfeit watermarked image $Z' = \{z_1', z_2', ..., z_N'\}$ such that $Z' \approx Z$;

Begin

For i=1 to N

Construction a block y_i such that $y_i \in Y$ and $z_i \approx y_i$;

Let $y_i = y_j^n$;

$\tilde{z}_i' = \tilde{y}_i$;

$\ddot{z}_{f(i,k_p,N)}' = \ddot{y}_{f(j,k_p,N_n)}^n$;

End

End.

Clearly, a successful synchronous counterfeiting attack is carried out by the following two steps: First, an attacker could construct a block y_i approximately to z_i such that y_i belongs to the watermarked blocks; second, the 7MSBs of z_i' and the LSB of $z_{f(i,k_p,N)}'$ is achieved by using the basic idea of synchronous replacement operation. That is why this kind of attack is called as synchronous counterfeiting attacks (SC attack).

The proposed attacks have the following differences compared to VQ attack [8]:

(1) The method of replacing image block is different: VQ attack replaces the whole image block; the proposed attacks divide image block into image content and watermark, which is carried out by modifying image block content and corresponding watermark synchronously;

(2) The requirement of watermarked image is different: VQ attack asks the sizes of watermarked images must be the same; the proposed attacks only need the same keys, no matter the size of watermarked images;

(3) The selectable range of approximate block is different: VQ attack only selects in the "the equivalence class" [8]; the proposed attacks could select in all watermarked blocks;

(4) The size of counterfeit image is different: VQ attack requires image to be counterfeited must be with the same size as watermarked images; the proposed attacks ask the size of counterfeit image could be the same as watermarked image or not.

Obviously, the selectable range of watermarked image and image blocks in the proposed SC attack could be much broader than VQ attack. Consequently, the quality of counterfeit watermarked image obtained by SC attack is better than by VQ attack under the same condition, in other words, when the same quality of counterfeit watermarked image is obtained by both attacks, the number of watermarked images used in SC attack is less than VQ attack.

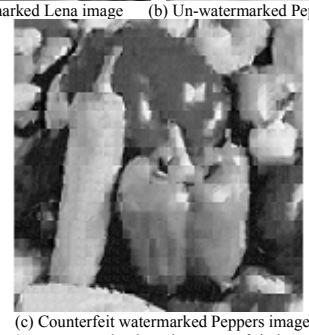
3.2 SC Attack Simulation

Either VQ attack or the proposed SC attack, the quality of counterfeit watermarked image both depends on the approximate degree of block $\tilde{z}_i'$ and $\tilde{z}_i$. Therefore, the more watermarked images are got, the better approximate blocks could be obtained, and the visual quality of counterfeited watermarked image will be better.

An example of SC attack on self-embedding watermarking algorithm [1, 7] is shown in figure 2. Figure 2(a) shows a legitimate watermarked image "Lena" with size of 256*256, and (b) is the original (un-watermarked) "Peppers" image. Distinctly, the contents of "Peppers" and "Lena" are very different. An approximation to "Peppers" image was then constructed using the watermarked "Lena" image, so as to contain a counterfeited watermark generated by self-embedding watermarking algorithm. The forged image is shown in figure 2(c). The PSNR between (b) and (c) is 22.8888.

Obviously, the perceptual quality of (c) is not very good. However, our aim here is just to demonstrate the possibility of SC attack. As our experimental results demonstrate, one image, unrelated to the watermarked image, could be counterfeited the watermarked image with reasonable quality by using other watermarked image.


(a) Watermarked Lena image (b) Un-watermarked Peppers image



(c) Counterfeit watermarked Peppers image

Fig. 2 Watermarked and counterfeit images

In general, as a consequence that the size of counterfeit watermarked image could not be same as watermarked images, the SC attack could be implemented to the image obtained by cutting out portions of watermarked image. If the cropping watermarked image, which is denoted as **Z**, is generated with block as a unit, forging watermarked image leads to excellent results in terms of perceptual quality. This is because each block in **Z** could be found a watermarked block with the same 7MSBs as itself. Accordingly, the 7MSBs of counterfeit watermarked image Y' obtained by SC attack is the same as **Z**, only LSB is different. That is to say the quality of counterfeit watermarked image and legitimate watermarked image are totally equal. Contrarily, the counterfeit quality of cropping image is not good if it was obtained without block as a unit.

Another example of such attack is shown in figure 3. A legitimate watermarked image "car-woman" with size of 256*392 is shown in figure 3(a). The cropping image car-woman (9:248, 89:384) with image block as a unit is called $\mathbf{Z}_1$; The cropping image $\mathbf{Z}_2$=car-woman(9:248,93:388), which is cropped without block as a unit. Counterfeit watermarked image of $\mathbf{Z}_1$ has been obtained by SC attack and shown in figure 3(b). The PSNR between (b) and $\mathbf{Z}_1$ is 51.1645 dB. The quality of (b) is superior. Figure 3(c) is the counterfeit watermarked image of $\mathbf{Z}_2$. The PSNR between (c) and $\mathbf{Z}_2$ is 22.6592 dB. Apparently, the perceptual quality of (c) is not good.

(a) Watermarked car-woman image

(b)Counterfeit watermarked image of $\mathbf{Z}_1$ (c) Counterfeit watermarked image of $\mathbf{Z}_2$

Fig. 3 Counterfeiting watermark in cropping image

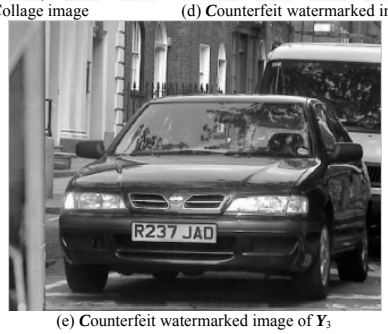
In addition, the size of watermarked images used to construct approximation blocks could not be same either. Therefore, the image achieved by cutting out portions of several watermarked images and combined them together using powerful image editing software could be forged watermark by SC attack, and the perceptual quality of counterfeit watermarked image is still good.

Taken the third example of such attack, we take two 256*256 girl images, shown in figure 4 (a) and (b), in which there exist watermark using the self-embedding technique in [1] or [7]. They were used in the construction of an approximation to figure 4(c). The collage image (c) is obtained by replacing the head of (a) with the head of (b) using powerful image editing software. The newly constructed image, containing a counterfeit watermark, is shown in figure 4(d). The PSNR between (c) and (d) is 34.8220 dB. Another collage image is obtained by pasting 1:32 columns of watermarked "Lena" in figure 2 (a) and 89:384 columns of watermarked "car-woman" in figure 3 (a), denoted as $\mathbf{Z}_3$. The counterfeit watermarked image of $\mathbf{Z}_3$ is shown in figure 4 (e). The PSNR between (e) and $\mathbf{Z}_3$ is 51.3276 dB.


(a) Watermarked image (b) Watermarked image

(c) Collage image

(d) Counterfeit watermarked image of (c)



(e) Counterfeit watermarked image of Y_3

Fig. 4 Counterfeiting watermark in collage image

As our experimental results demonstrate, the proposed SC attack has of several functions as follows: (1) Counterfeiting watermark in any un-watermarked image; (2) Changing the size of watermarked image randomly; (3) Generating the excellent quality of the counterfeit watermarked image by cropping and/or pasting watermarked images with image block as a unit. Apparently, the SC attack has strong ability to destroy self-embedding algorithm.

4. Conclusions

In this paper, we firstly point out the secure holes of the current self-embedding watermarking schemes, and the possibility of obtaining the offset according to watermarked image. Then we propose a novel synchronous attack on self-embedding watermarking algorithms. We have shown that the proposed SC attack may be possible to counterfeit watermark in an un-watermarked image without knowledge of the watermark insertion key, and the excellent quality of a counterfeited watermarked image is obtained using a few watermarked images under a certain condition. These facts strongly suggest that a poorly designed self-embedding watermarking scheme could be very susceptible to synchronous counterfeiting attacks proposed in this paper.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (Grant No.60572027), by the Outstanding Young Researchers Foundation of Sichuan Province (Grant No. 03ZQ026-033) and by National Defense Pre-study Foundation of China (Grant No. 5143080104QT2201).

Reference

[1] J.Fridrich, M.Goljan, "Images with Self-Correcting Capabilities", ICIP'99, Kobe, Japan, October 25-28, 1999.

[2]B.B.Zhu, M.D.Swanson, A.H.Tewfik, "When Seeing Isn't Believing", IEEE Signal Processing Magazine, March.2004, pp: 40-49.

[3] J.Fridrich, "Security of Fragile Authentication Watermarks with Localization", Proc. SPIE, Vol. 4675, Security and Watermarking of Multimedia Contents, San Jose, California, pp: 691-700, January, 2002.

[4] HE Hong-jie, ZHANG Jia-shu, TIAN Lei, "A Fragile Watermarking Scheme with Discrimination of Tampers on Image or Watermark", Chinese Journal of Electronics, Vol.33 No.9, pp:22-26,Sep. 2005.

[5] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," Proc. IEEE, vol. 87, no. 7, pp. 1167–1180, 1999.

[6] J.Fridrich, M.Goljan, "Protection of Digital images Using Self-Embedding", Symposium on Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, May 14, 1999.

[7] ZHANG Hong-bin, YANG Cheng. Tamper Detection and Self Recovery of Image Using Self-Embedding [J], Chinese Journal of Electronics, 2004.2 pp196-199)

[8] M.Holliman, N.Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes", IEEE Trans on Image Processing, pp: 432-441.2000.3(9).

[9] J.Fridrich, M.Goljan, N.Memon, "Cryptanalysis of the Yeung-Mintzer Fragile Watermarking Technique", Electronic Imaging, vol. 11, pp: 262-274, April 2002.

[10] Douglas R. Stinson. Cryptography Theory and Practice [M], 2003.2

Hongjie He was born in Henan, China, on Sep 4, 1971. She is pursuing the Ph.D degree in the the school of Information Science and Technology at Southwest Jiaotong University, Chengdu, Sichuan, China. Her research interests include digital watermarking, chaotic cryptography and image processing.

Jiashu Zhang received his B. S. and Ph. D degree from university of Electronic Science and Technology of China, in 1987 and 2001, respectively. In 2001 he joined the school of Information Science and Technology at Southwest Jiaotong University, Chengdu, Sichuan, China, where he is a professor of Signal and Information processing. His research interests focus on digital signal processing, information forensic and security, biometrics, and chaos theory with application to electronic engineering.