Adv Comput Math (2009) 31:267–281
DOI 10.1007/s10444-008-9096-1

A cryptographic watermarking technique for multimedia signals

Jian Ren

Received: 17 November 2007 / Accepted: 9 May 2008 /
Published online: 20 August 2008

Abstract Digital watermarking has been widely used in digital rights management and copyright protection. In this paper, new cryptographic watermark schemes are proposed. Compare to the existing watermarking techniques, our proposed watermark schemes combine both security and efficiency that none of the existing schemes can do. We first develop an algorithm to randomly generate the watermark indices based on the discrete logarithm problem (DLP) and the Fermat's little theorem. Then we embed watermark signal into the host image in both time domain and frequency domain at the indices. Our security analysis and simulation demonstrate that our proposed schemes can achieve excellent transparency and robustness under the major security attacks and common signal degradations. The novel approaches provided in this paper are ideal for general purpose commercial digital media copyright protection.

Keywords Data hiding · Information embedding · Watermarking · Cryptography

Mathematics Subject Classification (2000) 68W99

1 Introduction

Advances in computer networking and high speed computer processors have made duplication and distribution of multimedia data easy and virtually cost-

Communicated by Lixin Shen and Yuesheng Xu.

J. Ren (✉)
Department of Electrical and Computer Engineering,
Michigan State University, East Lansing, MI 48824, USA
e-mail: renjian@egr.msu.edu

less, and have also made copyright protection of digital media an ever urgent challenge. As an effective way for copyright protection, digital watermarking, a process which embeds (hides) a *watermark signal* in the *host signal* to be protected, has attracted more and more research attention [1–8]. The host signals are usually images, video or audio signals that are distributed in digital format. The watermark is desired to reside in the host signal permanently to provide copyright protection, license restriction, owner identification, content authentication, and traitor tracing [1].

The framework of watermarking can be depicted as in Fig. 1 [9, 10], in which a watermark signal w is embedded in the host signal h. The embedding is performed using a cryptographic key or the side information about the host signal. The resulted watermarked data may be subjected to both channel distortion and security attacks that attempt to remove or destroy the embedded watermark. Watermark extraction is the inverse process of watermark embedding. It can be performed either with or without access to the original signal, the later case is called *blind watermark extraction* [1, 11] and is especially useful when the host signal is not available. Compare to its non-blind counterpart, blind watermark extraction is generally more challenging and often has limited extraction capability, hence limits the amount of embedded data.

The digital watermark embedding system needs to satisfy the following three requirements:

1. *Perceptual transparency* or *fidelity* The embedded signal should be "hidden" within the host signal, and causes no serious degradation to its host.
2. *Robustness* The embedded signal should be robust to common degradations of the host signal, including benign processing and transmission degradation as well as malicious attacks.
3. *Security* The watermarking procedure should be performed in a way such that an unauthorized user is unable to detect the presence of the embedded signal, let alone remove the embedded data.

Existing watermark embedding algorithms can be roughly split into two different classes: time domain (also know as spatial domain) watermarking and frequency domain watermarking. In time domain approaches, the watermark information is embedded into the individual pixels selected under certain criteria. It only results in alterations to individual pixels. The "localized" alteration to the individual pixels tend to create highlighted spots that are

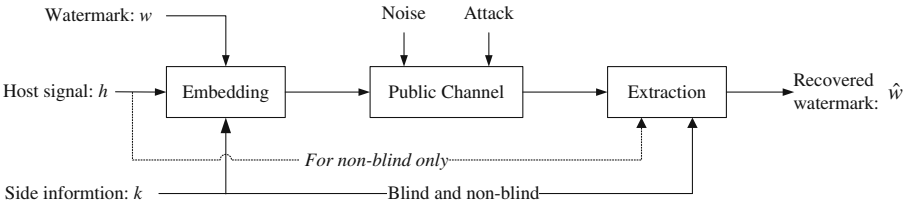

Fig. 1 Block diagram of watermark embedding and extraction processes

potentially detectable. While in frequency domain approaches, the watermark signal is embedded into the spectrum of the host image generated through the Fourier transform. The embedded watermark is therefore distributed to the entire host image following the inverse Fourier transform. As a result, the embedded watermark generally has a high transparency and is hard to be perceptually detected as long as the embedded watermark power is small relative to that of the host image.

Most of the existing watermarking schemes are based on signal processing techniques [1, 5, 7]. In [2], a watermarking scheme based on the RSA public-key scheme [12] was introduced for time domain. However, the scheme is very inefficient and cannot be applied to frequency domain straightforwardly. In an effort to increase the fidelity, robustness and security of the digital watermarking, in this paper we propose novel watermarking schemes based on the discrete logarithm problem (DLP) and the Fermat's little theorem for general purpose digital rights management. The watermark embedding includes two basic steps: the watermark index generation and the watermark embedding. Our proposed approach can be applied to both time domain and frequency domain with high transparency and robustness under common signal degradations such as Gaussian noise and random multiplicative noise. The proposed watermark techniques are also secure under major watermark attacks such as cropping and least significant bit (LSB) destruction.

The rest of this paper is organized as follows. In Section 2, cryptographic watermark index generation based on discrete logarithm is introduced. Section 3 and Section 4 are devoted to time domain and frequency domain digital image watermarking approaches, respectively. Security analysis is provided in Section 5. Simulation results are presented in Section 6, and we conclude in Section 8.

2 Preliminary

In this section, we will introduce some preliminary for this paper.

One-way function *One-way function* plays an important role in modern cryptography. Mathematically, a one-way function can be characterized as a relation $f : X \to Y$ that is easy/efficient to calculate, but very difficult/inefficient or even impossible to inverse. More specifically,

1. For any given $x \in X$, $y = f(x)$ can be efficiently computed (or computed in polynomial time).
2. Given $y = f(x)$ for some $x \in X$, it is computationally difficult, or infeasible to find x.

Discrete Logarithm Problem (DLP) Let p be a prime number, and α a primitive element of $\mathbb{Z}_p$ (i.e., a generator of $\mathbb{Z}_p^*$), where $\mathbb{Z}_p$ is the integer field modulo p. The function $f_{\alpha,p} : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ defined by $f_{\alpha,p}(x) = \alpha^x \bmod p$

is a permutation on $\mathbb{Z}_p^*$ that is computable in polynomial time. The *discrete logarithm problem*, with parameters α and p, consists in finding for each y in $\mathbb{Z}_p^*$ the index x in $\mathbb{Z}_p^*$ such that $\alpha^x \bmod p = y$. A (probabilistic) procedure $\mathbf{p}[\alpha, p, y]$ solves the discrete logarithm problem if for all primes p, for all generators α of $\mathbb{Z}_p^*$, and for all y in $\mathbb{Z}_p^*$, $\mathbf{p}[\alpha, p, y] = x$ in $\mathbb{Z}_p^*$ such that $\alpha^x \bmod p = y$.

The DLP in a cryptographic setting is that finding discrete logarithms is very difficult for large integers, but the inverse operation can be computed efficiently. In other word, exponential operation is a one-way function in $\mathbb{Z}_p^*$ when p is a large prime.

For prime number p, we need to introduce the following well-known Fermat's little theorem.

Lemma 1 (Fermat's little theorem [13]) *Let p be a prime number. If a is an integer that is not divisible by p, then $a^{p-1} = 1 \bmod p$.*

3 Watermark index generation

The following theorem provides the theoretical foundation for watermark embedding and extracting. We hope to embed the watermark signal at the locations calculated from the following lemma. Our goal is to embed the watermark signal into the entire host image secretly, evenly and randomly.

Theorem 1 *Let p be a prime and α a primitive element in $\mathbb{Z}_p$. If $b \in \mathbb{Z}_p$ is a randomly selected integer such that $(p-1) \nmid bi$ for any $1 \le i \le l \le p-2$, then for any $a \in \mathbb{Z}_p$,*

$$\alpha^a, \alpha^{a+b}, \alpha^{a+2b}, \cdots, \alpha^{a+(l-1)} \bmod p, \tag{1}$$

are all distinct.

Proof Assume that for some $i, j, 1 \le i < j \le l$ we have $\alpha^{a+ib} = \alpha^{a+jb} \bmod p$, which is equivalent to $\alpha^{b(j-i)} = 1 \bmod p$. According to the Fermat's little theorem, we also have $\alpha^{p-1} = 1 \bmod p$. Since α is a primitive element in $\mathbb{Z}_p^*$, and $\alpha, \alpha^2, \cdots, \alpha^{p-1} = 1 \bmod p$ are all different. Therefore, $p-1$ is the order of α. Hence, we have $(p-1) \mid b(j-i)$. This contradicts to the assumption that $(p-1) \nmid bi$ for any $1 \le i < l$. Therefore, $\alpha^a, \alpha^{a+b}, \cdots, \alpha^{a+(l-1)b} \bmod p$ are all distinct. □

Now suppose we want to ensure that the entire sequence generated in (1) is smaller than a predetermined integer v, it is natural to substitute $\alpha^{a+ib} \bmod p$ with $(\alpha^{a+ib} \bmod p) \bmod (v+1)$. However, the problem for $\bmod(v+1)$ operation is that under this operation, two different terms in the sequence may become the same. This can be seen in the following example.

Example 1 Suppose we have $p = 7, v = 4, \alpha = 3, a = b = 1$, it can be verified very easily that our selection satisfies all the requirements of Theorem 1.

In addition, we have $\alpha^{a+b} = 3^2 = 2 \bmod p$, $\alpha^{a+2b} = 3^3 = 6 \bmod p$. Therefore, $\alpha^{a+b} \neq \alpha^{a+2b} \bmod p$. However, it is easy to verify that $\alpha^{a+b} \neq \alpha^{a+2b} = 2 \bmod p \bmod v$.

To solve the problem, we substitute $\alpha^{a+ib} \bmod p$ in (1) with $T(\alpha^{a+ib})$ defined as follows:

$$T(\alpha^{a+ib}) = \min\{k \mid k \geq (\alpha^{a+ib} \bmod p) \bmod (v+1), \\ k \neq (\alpha^{a+jb} \bmod p) \bmod (v+1) \text{ for } j < i\}, \tag{2}$$

where "$\geq$" is a relation defined on $\{1, 2, \cdots, v\}$. $x \geq y$ if and only if x is next to y in the clockwise direction as shown in Fig. 2. As an example, we can see that $1 \geq v$.

The intuitive explanation for this transform is that if $(\alpha^{a+ib} \bmod p) \bmod (v+1)$ is different from any $T(\alpha^{a+jb})$ for $1 \leq j < i$, then we define $T(\alpha^{a+ib}) = (\alpha^{a+ib} \bmod p) \bmod (v+1)$. However, if $(\alpha^{a+ib} \bmod p) \bmod (v+1)$ is the same as $(\alpha^{a+jb} \bmod p) \bmod (v+1)$ for some $j < i$, then we define $T(\alpha^{a+jb})$ to be the immediate next element in the clockwise direction that is different from any of the $T(\alpha^{a+jb})$ for $j < i$, shown in Fig. 2.

Based on this modification, we generate the following modified sequence of watermark indices

$$r = \left(T(\alpha^a), T(\alpha^{a+ib}), \cdots, T(\alpha^{a+(v-1)b})\right). \tag{3}$$

The following lemma characterizes the number of options in selecting the parameter α for (3).

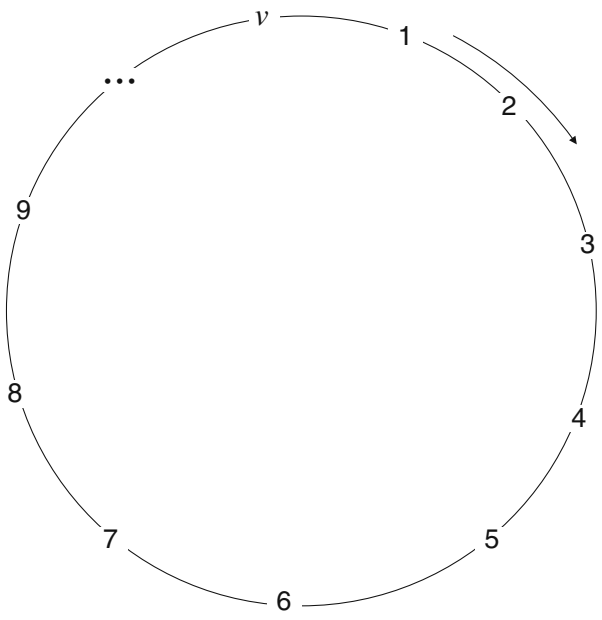

Fig. 2 Illustration of the transform for (2)

Lemma 2 [13] *$\mathbb{Z}_p$ contains $\phi(p-1)$ primitive elements, where $\phi(n)$ is the Euler's function defined as the number of integers k with $1 \le k \le n$ and* $\gcd(k, n) = 1$. *Moreover, if $p - 1 = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ is the prime factor decomposition of $p-1$, then $\phi(p-1) = (p-1)\left(1 - \frac{1}{p_1}\right)\cdots\left(1 - \frac{1}{p_k}\right)$.*

Example 2 The number of primitive elements in $\mathbb{Z}_{491}$ is $\phi(490) = 168$. The number of primitive elements in $\mathbb{Z}_{769}$ is $\phi(768) = 256$.

More discussion on the Euler's function can be found in [13]. From the above example, we can see that $\phi(n)$ can be quite large when n is large.

The watermark index generation algorithm includes two major steps:

1. Let l in Theorem 1 be the number of bits to be embedded. Select p, α, a and b according to Theorem 1. Lemma 2 ensures the existence of α in large number, as there are altogether $\phi(p-1)$ primitive elements in $\mathbb{Z}_p$.
2. The watermark indices can then be generated according to (2) and (3).

4 Time domain image watermarking

In time domain watermarking, the watermark signal is embedded into individual pixels. If the watermarked individual pixel is distinct apart from the neighboring pixels, it may result in highlighted spot and make the watermark perceptually detectable. It is, therefore, natural to embed watermark signals only to the locations with pixel values larger than a given threshold. However, threshold-based watermarking may severely complicate the watermark extraction process if blind watermark extraction is expected, as even the manipulation of a single pixel with watermark embedded could prevent the whole watermark from being recovered blindly, i.e., watermark is being recovered without access to host image. In the following, watermark embedding both with and without threshold will be discussed.

Let h be a host image indexed as $h_x \times h_y \times h_z$, where $h_x \times h_y$ is the image size, $h_z = 1$ if h is an 8-bit gray-scale image, and $h_z = 3$ if h is a 24-bit truecolor (RGB) image, which corresponds to the three color planes.

In this paper, for simplicity, each 24-bit truecolor image is viewed as three concatenated gray-scale images, and each gray-scale pixel is represented as an 8-bit array. Let $l_h = h_x h_y h_z$, the whole image h can be represented using a bit array of length $L_h = 8l_h$. Let w be the watermark image indexed by $w_x \times w_y \times w_z$ and $l_w = w_x w_y w_z$, then w can be represented as a binary array of length $L_w = 8l_w$, which is still denoted as w.

4.1 Embedding without threshold

As is well known, in digital watermarking, there is always a trade-off between *robustness* and *fidelity*. In time domain, if the watermark is embedded at the least significant bits (LSBs), the watermarked image generally maintains a high

fidelity but sacrifices the robustness. On the other hand, if the watermark is embedded in the most significant bits (MSBs), then the opposite is true. That is the watermark is generally robust but sacrifices the fidelity.

To balance between robustness and fidelity, in this section, only the four middle bits of each 8-bit gray-scale pixel will be selected as the source for possibly watermark embedding. Let $b_0b_1b_2b_3b_4b_5b_6b_7$ be the binary representation of a gray-scale pixel, then the watermark will be embedded only among the bits $b_2b_3b_4b_5$.

The host image is first reshaped to an integer array of size $l_h = h_xh_yh_z$ (the reshape method is not critical as long as it is consistent for both the embedding and the extraction processes), where the integers are in the range of 0 to 255. The integer array is then converted to a binary matrix H of size $l_h \times 8$.

Let H_s be the matrix generated by extracting the four middle columns in matrix H. H_s can then be reshaped to a bit array h_s of length $L_s = 4l_h$, where the watermark w of size $L_w = 8l_w < L_s$ will be embedded, here $l_w = w_xw_yw_z$ as indicated before.

The ith watermark bit $w(i)$ will be used to replace the $(r(i), c(i))$-th entry of matrix H_s, where the row number $r(i)$ is calculated from (3), $v = L_w$, the column number $c(i) = (i \bmod 4) + 1$. This approach ensures that the watermark information is embedded evenly into the four middle bits. More specifically, the embedding algorithm can be summarized as follows:

Algorithm 1—Time domain watermark embedding

(1.1) Convert the host image h of size $h_x \times h_y \times h_z$ to a binary matrix H of size $l_h \times 8$.
(1.2) Select the four middle columns from H to form a submatrix H_s. The size of H_s is $l_h \times 4$.
(1.3) Select a large prime p, a primitive element α mod p and the secret keys a, b according to Theorem 1.
(1.4) Calculate the watermark index sequence r according to (3), and let $c = [1, 2, 3, 4, 1, 2, 3, 4, \cdots, 1, 2, 3, 4]$, here c is of the same size as r.
(1.5) Convert the watermark image w of size $w_x \times w_y \times w_z$ to a bit-string of size L_w. The generated bit-string will still be denoted as w.
(1.6) Substitute each bit $H_s(r(i), c(i))$ with the ith watermark bit $w(i)$, for $1 \leq i \leq L_w$.
(1.7) Reverse step (1.1) to obtain the watermarked image.

The watermark extraction process follows directly from the embedding algorithm described above.

Remark 1 To achieve good fidelity and robustness, it is essential to limit the ratio between the watermark size and the host image size to a certain range.

An excessive large watermark could severally distort the host image and make the watermark perceptually detectable. Define the embedding ratio (EBR) as

$$EBR = \frac{L_w}{L_h} = \frac{2l_w}{l_h}.$$

Simulation results demonstrate that as long as the EBR is limited to around 5%, the watermarked image generally maintains high perceptual transparency and robustness.

4.2 Time domain watermark embedding with threshold

Algorithm 1 can be slightly modified for watermark embedding with a given threshold, so that watermark signal is only embedded at the pixels with values larger than the threshold.

Algorithm 2—Watermark embedding at the pixels above the threshold T

(2.1) Convert the host image h of size $h_x \times h_y \times h_z$ to an integer string of size l_h, where each integer is within [0,255].
(2.2) Find all the integers larger than the threshold T and let t be the total number of such integers.
(2.3) Convert the selected integer sequence to a binary matrix H of size $t \times 8$.
(2.4) Select the 4 middle columns from H, denote the obtained $t \times 4$ matrix as H_s.
(2.5) Select a large prime p, a primitive element α mod p and the secret keys a, b according to Theorem 1.
(2.6) Perform steps (1.4)–(1.7) in Algorithm 1.

Remark 2 For security consideration, T must be selected to ensure that t is significantly larger than l_w.

Remark 3 Theoretically, embedding with threshold can assure high transparency and robustness by limiting the maximum pixel variation of the host image in a controlled manner. However, the corresponding watermark extraction process generally requires access to the host image, because the subset of the pixels above the threshold and the sequential orders may be varied by even a slight manipulation of the watermarked image.

5 Frequency domain watermarking

In this section, we apply the proposed random embedding scheme to frequency domain watermarking. In time domain watermark embedding, we simply *replace* the individual host image bits with the watermark bits. A similar watermark embedding approach in frequency domain may cause dramatic quality degradation to the host image, which could also render the watermark from being successfully recovered [7]. Therefore, instead of replacing the original

"host" image element, we propose to first scale the watermark signal down by multiplying a small factor, then *add* the watermark signal to the host image in frequency domain.

To embed watermark evenly to the entire host image, prior to applying the FFT (fast Fourier transform) operation, the host image is first reshaped to a long integer array as in the time domain algorithms. The algorithm is described as follows:

Algorithm 3—Frequency domain watermark embedding

(3.1) Convert the host image and the watermark image to integer arrays of size l_h and l_w, respectively, where each integer is within [0, 255].
(3.2) Perform FFT for both h and w, and let $\mathcal{F}h \leftarrow \text{FFT}(h)$, $\mathcal{F}w \leftarrow \text{FFT}(w)$ be the FFT of h and w, respectively.
(3.3) Select a large prime p, the primitive element α mod p and the secret keys a, b as described in Theorem 1.
(3.4) Calculate the embedding indices according to (3).
(3.5) Update sequence $\mathcal{F}h$ by adding sequence $\mathcal{F}w$ multiplied by a very small scaler to it at indices generated in step (3.4).
(3.6) Apply the inverse FFT (IFFT) to the updated $\mathcal{F}h$. Let $h_w \leftarrow IFFT(\mathcal{F}h)$.
(3.7) Normalize h_w through three simple processes: a) substitute h_w with the complex modulus (magnitude) of h_w; b) converts the array h_w into unsigned 8-bit integer; and c) reshape h_w to the dimension of the original host image.

In frequency domain watermarking, the embedded watermark will have an impact on all the pixels in the host image through the inverse Fourier Transform. As a result, as long as the embedded watermark is small comparing to the host image, it is hard for the embedded watermark to be perceptually detected. That is, the watermark has excellent transparency. It is very likely that after the IFFT operation in step (3.6), the watermarked host image may no longer be real numbers or integers. Part of the reason is that watermark embedding itself will create distortion to the original host image.

6 Security analysis

The security of the proposed watermark embedding relies on the discrete logarithm problem (DLP) with parameters: α, a, b. In the proposed scheme, α can be any of the $\phi(p-1)$ different numbers, where p is a large prime, a can be any number from 1 to p, and b has $(p-1)/L_w$ different selections according to Theorem 1.

To determine whether a randomly selected triple (α, a, b) is the right secret key, in time domain one needs to extract all the L_w bits from the watermarked image, which requires L_w discrete logarithms. To fully determine the secret keys for a embedded watermark in a particular image, it requires

altogether $\phi(p-1)(p-1)(p-1)/L_w L_w = (p-1)^2\phi(p-1)$ different discrete logarithms and the corresponding image reconstruction process to fully recover the embedded watermark. As an example, for a 512×512 24-bit truecolor (RGB) image, it requires $124514889179848506880 \approx 1.2451 \times 10^{20}$ discrete logarithms. Without the secret key, it is computationally infeasible to extract the embedded watermark. While in frequency domain, in addition to the complexity required for the time domain, one also needs to determine how to separate the watermark signal from the watermarked spectrum for each index. Therefore, watermark extraction in frequency domain is much more difficult for the adversary than that in the time domain scenario. Overall, the proposed watermark embedding algorithms have high security under known attacks.

7 Simulation results

In this section, performance analysis and simulation results are provided to demonstrate the robustness and fidelity of the proposed approaches. It is inevitable that the watermark will create some distortion to the host image. To evaluate the quality of the watermark image and measure the distortion to the host image, we will introduce the concept of *mean-square error* (MSE) and *peak signal-to-noise* (PSNR) [2, 3, 14, 15] below. MSE is defined as

$$MSE = \frac{1}{l_w} \sum_{k \in \mathcal{S}} |\alpha_k - \beta_k|^2,$$

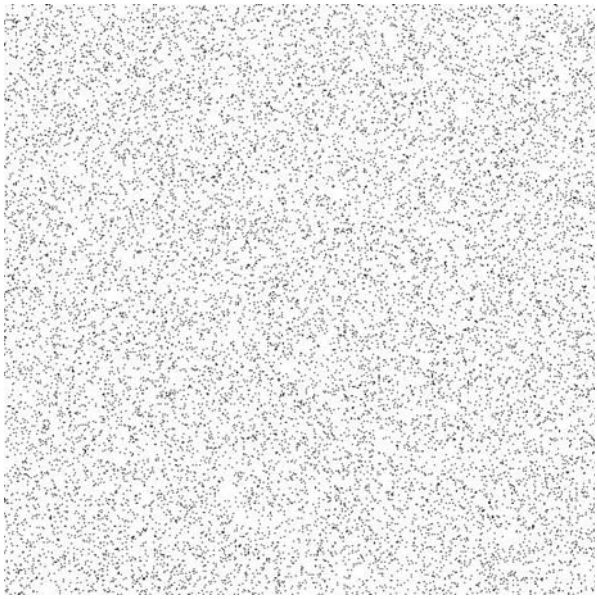

Fig. 3 Distribution of watermark information

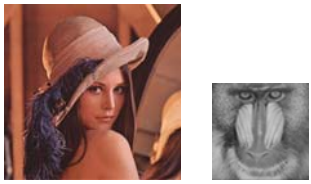

Fig. 4 Time domain watermarked image and the original watermark: **a** Watermarked image (MSE = 7.07, PSNR = 44.41), **b** watermark

where l_w is the total number of pixels in the host image, $\mathcal{S}$ is the set of indices of all the pixels, α_s and β_s are the pixels of the host image and the watermarked images, respectively. PSNR is defined as

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ (dB)}.$$

For time domain watermarking, embedding watermark signals only at the four middle bits can provide an ideal balance between transparency and robustness. For example, tamper of the two LSBs will not affect the watermark signal, while keeping the two most significant bits unaltered caps the maximal distortion that the watermark may add to the host image. Random distribution of the watermark signals, shown in Fig. 3, assures that the watermark is distributed to the entire host image. This property also guarantees the robustness of embedded watermark under security attacks and image cropping [16, 17]. In the following examples, the watermark used was the familiar "mandrill" image often employed in image processing research.

In Fig. 4, we show the watermarked image and the recovered watermark signal in the time domain. Figure 5 simulates the extracted watermarks under different LSBs destruction of the watermarked image. Figure 6 presents the extracted watermark under different image cropping. Figure 7 demonstrates the extracted watermark when noise is added to the watermarked signal. No threshold is used for all these simulations.

Figure 8 is similar to Figs. 4, 6 and Fig. 7, except that a pre-selected threshold is used in watermark embedding and extraction. Our simulations results

Fig. 5 Time domain example 1 (without threshold)—watermarked images with LSB destructions and the recovered watermarks: **a** Three LSBs destroyed (MSE = 24.64, PSNR = 38.99), **b** four LSBs destroyed (MSE = 84.43, PSNR = 33.64), **c** five LSBs destroyed (MSE = 324.54, PSNR = 27.79)

Fig. 6 Time domain example 2 (without threshold)—cropped watermarked images and the recovered watermarks. *d* Cropped the lower 1/3 (MSE = 3,531.6, PSNR = 17.42), *e* cropped the right 1/3 (MSE=1,330.7, PSNR=21.66), *f* cropped the 50 outside pixels (MSE = 4,552.6, PSNR = 16.32)

Fig. 7 Time domain example 3 (without threshold)—noise distorted watermarked images and the recovered watermarks: *g* Add zero mean Gaussian noise with variance 0.0001 (MSE = 13.65, PSNR = 41.55), *h* added "salt and pepper" noise with density 0.2 (MSE = 4,050.3, PSNR = 16.83), *i* adds uniformly distributed random multiplicative noise with zero mean and variance 0.001 (MSE = 30.39, PSNR = 38.07)

Fig. 8 Time domain example 4 (with threshold $T = 32$)—noise distorted watermarked images and the recovered watermarks: **a** Watermarked image and the embedded watermark (MSE = 6.96, PSNR = 44.48), **b** cropped the 50 outside pixels (MSE = 4,552.3, PSNR = 16.32), **c** add zero mean Gaussian noise with variance 0.0001 (MSE = 13.55, PSNR = 41.58)

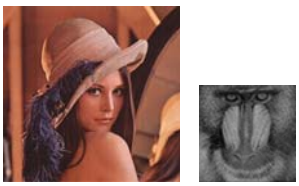

Fig. 9 Frequency domain example 1—watermarked image and the recovered watermark: MSE = 0.8394, PSNR = 48.8912

Fig. 10 Frequency domain example 2—watermarked images with LSB destructions and the recovered watermarks: **a** Three LSBs destroyed (MSE = 23.4455, PSNR = 34.4302), **b** four LSBs destroyed (MSE = 87.8464, PSNR = 28.6936), **c** five LSBs destroyed the (MSE = 339.0723 PSNR = 22.8279)

Fig. 11 Frequency domain example 3—cropped watermarked images and the extracted watermarks: *d* Cropped the lower 1/3 (MSE = 0.6111, PSNR = 50.2696), *e* cropped the right 1/3 (MSE = 0.6142, PSNR = 50.2479), *f* cropped the 50 outside pixels (MSE = 0.5424, PSNR = 50.7873)

Fig. 12 Frequency domain example 4—noise distorted watermarked images and the recovered watermarks: *d* Add zero mean Gaussian noise with variance 0.0001 (MSE = 346.5164, PSNR = 22.7336), *e* added "salt and pepper" noise with density 0.002 (MSE = 370.7324, PSNR = 22.4402), *f* adds uniformly distributed random multiplicative noise with zero mean and variance 0.001 (MSE = 353.2917, PSNR = 22.6495)

Fig. 13 Frequency domain example 5—compressed watermarked image and the recovered watermark: *g* Applied JPEG image compression with quality parameter 85% (MSE = 22.6896, PSNR = 34.5725)

demonstrate that the proposed time domain watermark approaches are robust under LSB destruction, several image cropping, and watermarked image distortion caused by Gaussian noise, "salt and pepper" noise and speckle (multiplicative) noise.

We also provide simulation results for frequency domain watermark. Figure 9 gives the watermarked image and the extracted watermark without security attack and signal noise presented. Figure 10 presents the extracted watermark under different LSBs destruction of the watermarked image. In Fig. 11, we present the recovered watermark when part of the watermarked image is replaced with the host image. Figure 12 shows the extracted watermark in frequency domain, where the watermarked signal is distorted due to signal noise. Our final simulation shows the extracted watermark when the watermarked signal is compressed with quality parameter 85%. Our simulations results demonstrate that the proposed watermark is robust under LSB destruction, severe image cropping, JPEG compression and signal distortions caused by Gaussian noise, "salt and pepper" noise and speckle (multiplicative) noise.

We observed that time domain watermarking and frequency domain watermarking each has its own advantages: time domain watermarking tends to be robust under large block image cropping with random fill for the cropped areas, as shown in Fig. 6, while due to the built-in frequency diversity, frequency domain watermarking shows high robustness under JPEG image compression, as shown in Fig. 13. The major reason for the robustness of the watermark schemes is because the watermark signal is embedded into the host image randomly and in the entire domain.

8 Conclusions

In this paper, three simple but efficient cryptographic watermark embedding schemes were developed in both time domain and frequency domain. In addition to excellent transparency and robustness under various degradations, the proposed approaches were also shown to be secure under major watermark attacks. Furthermore, it is also observed that affected by locality and globality, time domain and frequency domain digital watermarking approaches each has its own advantages.

References

1. Swanson, M., Kobayashi, M., Tewfik, A.: Multimedia data embedding and watermarking technologies. Proc. IEEE **86**(6), 1064–1087 (June 1998)
2. Hwang, M.-S., Chang, C.-C., Hwang, K.-F.: A watermarking technique based on one-way hash functions. IEEE Trans. Consumer Electron. **45**(2), 286–294 (May 1999)
3. Chen, B., Gornell, G.W.: Quantization index modulation: a class of probably good methods for digital watermarking and information embedding. IEEE Trans. Inf. Theory **47**(4), 1423–1443 (May 2001)
4. Malvar, H.S., Florêncio, D.A.F.: Improved spread spectrum: a new modulation technique for robust watermarking. IEEE Trans. Signal Proc. **51**(4), 898–905 (April 2003)

5. Xie, L., Arce, G.R.: A class of authentication digital watermarks for secure multimedia communication. IEEE Trans. Image Proc. **10**(11), 1754–1764 (November 2001)
6. Tang, C.-W., Hang, H.-M.: A feature-based robust digital image watermarking scheme. IEEE Trans. Signal Proc. **51**(4), 950–959 (April 2003)
7. Cox, I., Kilian, J., Leighton, F., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process **6**(12), 1673–1687 (December 1997)
8. Wang, S.-H., Lin, Y.-P.: Wavelet tree quantization for copyright protection watermarking. IEEE Trans. Image Process **13**(2), 154–165 (February 2004)
9. Moulin, P., O'Sullivan, J.: Information-theoretic analysis of information hiding. IEEE Trans. Inf. Theory **49**(3), 563–593 (March 2003)
10. Moulin, P., Mihçak, M.: The parallel-Gaussian watermarking game. IEEE Trans. Inf. Theory **50**(2), 272–289 (September 2004)
11. Craver, S., Memon, N., Yeo, B., Yeung, M.: Can invisible watermarks resolve rightful ownerships? IBM Tech. Rep. RC 20509 (July 1996)
12. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. Assoc. Comp. Mach. **21**(2), 120–126 (1978)
13. Lidl, R., Niederreiter, H.: Finite Fields. Cambridge University Press (2000)
14. Pérez-González, P., Balado, F., Martin, J.R.H.: Performance analysis of existing and new methods for data hiding with known-host information in additive channels. IEEE Trans. Signal Proc. **51**(4), 960–980 (April 2003)
15. Moulin, P., Mihçak, M.: A framework of evaluating the data-hiding capacity of image sources. IEEE Trans. Image Process **11**(9), 1029–1042 (September 2002)
16. Chandramouli, R., Memon, N.D.: On sequential watermark detection. IEEE Trans. Signal Proc. **51**(4), 1034–1044 (2003)
17. Kirovski, D., Petitcolas, F.A.: Blind pattern matching attack on watermarking systems. IEEE Trans. Signal Proc. **51**(4), 1045–1053 (April 2003)