Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 1, 93–104
DOI: 10.2478/v10006-008-0009-8

M-ARY PHASE MODULATION FOR DIGITAL WATERMARKING

YONGQING XIN, MIROSŁAW PAWLAK

Department of Electrical and Computer Engineering
University of Manitoba, Winnipeg
Manitoba, Canada R3T 5V6
e-mail: pawlak@ee.umanitoba.ca

In spread spectrum based watermarking schemes, it is a challenging task to embed multiple bits of information into the host signal. M-ary modulation has been proposed as an effective approach to multibit watermarking. It has been proved that an M-ary modulation based watermarking system outperforms significantly a binary modulation based watermarking system. However, in the existing M-ary modulation based algorithms, the value of M is restricted to be less than 256, because as M increases, the computation workload for data extraction advances exponentially. In this paper, we propose an efficient M-ary modulation scheme, i.e., M-ary phase modulation, which reduces the computation in data extraction to a very low level. With this scheme, it is practical to implement an M-ary modulation based algorithm with a high value of M, e.g., $M = 2^{20}$. This is significant for a watermarking system, because it can either greatly increase the data capacity of a watermark given the necessary watermark robustness, or considerably improve the watermark robustness given the amount of information of the watermark. The superiority of the proposed scheme is verified by simulation results.

Keywords: Multibit watermarking, M-ary phase modulation, watermark robustness, data capacity.

1. Introduction

Watermarking systems based on the spread spectrum technique (Cox *et al.*, 1997; Cox *et al.*, 2001) have been prevalent due to their distinguishing characteristics such as good security and robustness performance. However, some fundamental issues on spread spectrum based watermarking methods are still open to investigation. For instance, a challenging task in the design of a spread spectrum based watermarking system is to increase the amount of hidden data, given a fixed level of signal fidelity and watermark robustness. This is our concern in this paper.

Let us first look at how a 1-bit watermarking system works. Assume that $\mathbf{X} = (X[1], \ldots, X[L])$ is a vector of signal features selected for watermarking, which can be original signal samples, or coefficients of some transform, such as DCT, DFT, and DWT, and the message to embed is a binary digit $m \in \{0, 1\}$. For the embedding of the message bit m, we first generate two independent i.i.d. pseudonoise sequences (PNSs) $\mathbf{W}_0 = (W_0[1], \ldots, W_0[L])$ and $\mathbf{W}_1 = (W_1[1], \ldots, W_1[L])$[1] with a key K, where $W_j[i] \sim \mathcal{N}(0, 1), j = 0, 1$ and $i = 1, \ldots, L$. The basic idea is that we use $\mathbf{W}_0$ and $\mathbf{W}_1$ to represent '0' and '1', respectively. $\mathbf{W}_\mathrm{m}$, the PNS used to modify the host signal, is either $\mathbf{W}_0$ or $\mathbf{W}_1$, depending on the bit value to be embedded:

$$\mathbf{W}_\mathrm{m} = \begin{cases} \mathbf{W}_0 & \text{if} \quad m = 0, \\ \mathbf{W}_1 & \text{if} \quad m = 1. \end{cases} \tag{1}$$

Then the watermarked signal is obtained as an additive mixture of $\mathbf{X}$ and $\mathbf{W}_\mathrm{m}$,

$$\tilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_\mathrm{m}, \tag{2}$$

where a is a constant watermark strength factor.

For watermark extraction from $\tilde{\mathbf{X}}$, $\mathbf{W}_0$ and $\mathbf{W}_1$ are re-generated with the same key K. Afterwards a certain detector $\mathcal{S}(\cdot)$ is invoked for the calculation of detection statistics between $\tilde{\mathbf{X}}$ and both $\mathbf{W}_0$ and $\mathbf{W}_1$, respectively. The embedded bit is estimated based on the following de-

[1] One can set $\mathbf{W}_1 = -\mathbf{W}_0$ to obtain a bi-orthogonal PNS set, which gives slightly better performance. For simplicity of presentation, bi-orthogonal PNSs are not discussed in this paper.

cision rule:

$$\hat{m} = \begin{cases} 0 & \text{if } \mathcal{S}(\tilde{\mathbf{X}}, \mathbf{W}_0) > \mathcal{S}(\tilde{\mathbf{X}}, \mathbf{W}_1) \\ & \text{and } \mathcal{S}(\tilde{\mathbf{X}}, \mathbf{W}_0) > T_s, \\ 1 & \text{if } \mathcal{S}(\tilde{\mathbf{X}}, \mathbf{W}_1) > \mathcal{S}(\tilde{\mathbf{X}}, \mathbf{W}_0) \\ & \text{and } \mathcal{S}(\tilde{\mathbf{X}}, \mathbf{W}_1) > T_s, \\ \text{none} & \text{if } \max\{\mathcal{S}(\tilde{\mathbf{X}}, \mathbf{W}_0), \mathcal{S}(\tilde{\mathbf{X}}, \mathbf{W}_1)\} < T_s, \end{cases} \tag{3}$$

where T_s is a pre-determined threshold for a required false alarm rate. If $\mathbf{X}$ follows Gaussian distribution, the watermark detector $\mathcal{S}(\cdot)$ can be implemented with a linear correlator,

$$\mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}) = \frac{1}{L} \sum_{i=1}^{L} \tilde{X}[i] W[i], \tag{4}$$

where L is the number of elements in the vector $\mathbf{W}$. If $\mathbf{X}$ is not Gaussian distributed, one can employ a certain optimal method (Zeng and Liu, 1999; Hernandez *et al.*, 2000; Cheng and Huang, 2001; Nikolaidis and Pitas, 2003).

In this paper, we focus on the problem of multibit watermarking based on the spread spectrum technique. We consider the situation of blind watermark extraction/decoding, in which the host signal serves as noise. As pointed out in (Cox *et al.*, 2001), based on a 1-bit watermark, one can design a multibit watermark by employing signal multiplexing techniques originating from communication theories (Wilson, 1996; Proakis, 2000). The most straightforward methods are based on feature space division, such as time/space/frequency division multiple access (TDMA/SDMA/FDMA). These intuitive approaches have the advantage of easy implementation, but the watermark embedded in this way is vulnerable to signal cropping and/or signal filtering. Another disadvantage is that different feature groups may have different sensitivities to distortions, thus leading to uneven watermark robustness. To overcome the limitations of feature division based techniques, code division multiple access (CDMA) can be considered for N-bit watermarking. The idea is to use the same feature vector many times; each time a separate message symbol is embedded as a layer of noise (from the perspective of the host signal). Based on TDMA/SDMA/FDMA/CDMA, one can embed a multibit watermark with multiple PNSs. Multibit watermarking systems based on these techniques have one disadvantage in common: achieving payload amount at the cost of either watermark robustness or signal fidelity.

M-ary modulation, on the other hand, can take advantage of only one PNS to communicate a multibit message. M-ary modulation has been utilized in communication theory (Wilson, 1996; Proakis, 2000) for some time, and recently was applied to digital watermarking by several authors (O'Ruanaidh and Pun, 1998; Kutter, 1999; Cox *et al.*, 2001; Trappe *et al.*, 2003). It was shown that the performance of a watermarking system can be considerably improved by M-ary modulation (Kutter, 1999). However, in practice, this advantage is limited by the computational cost in message decoding. In this paper, we show that with a proper choice of reference patterns, this limitation can be considerably mitigated.

This paper is organized as follows: In Section 2, we briefly introduce the concept of M-ary modulation based multibit watermarking and the limitation imposed by the existing decoding methods. In Section 3, we focus on an efficient implementation of M-ary modulation, i.e., M-ary phase modulation by means of circular versions of a PNS. The error performance of M-ary modulation based watermarks is derived in Section 4. In Section 5, a practical design of a multibit watermarking system based on M-ary phase modulation and its empirical performance under some common attacks are presented. Finally we conclude the paper in Section 6.

2. Conventional *M*-ary modulation based multibit watermarking

Suppose we have a feature vector $\mathbf{X} = (X[1], \ldots, X[L])$, which can be DCT, DWT or other transform domain coefficients of a host signal. Our objective is to modify $\mathbf{X}$ slightly with a same length watermark sequence $\mathbf{W} = (W[1], \ldots, W[L])$ to produce a watermarked feature vector $\tilde{\mathbf{X}} = (\tilde{X}[1], \ldots, \tilde{X}[L])$, through an embedding function $\mathcal{E}$, such that N message bits are hidden in $\tilde{\mathbf{X}}$, and later can be extracted from $\tilde{\mathbf{X}}$ without access to $\mathbf{X}$. An effective method is to use an M-ary modulation technique based on PNSs. A group of $M = 2^N$ pseudonoise patterns $\{\mathbf{W}_0, \ldots, \mathbf{W}_{M-1}\}$ are generated independently with a secret key K, each of which is an L-element i.i.d. sequence, following Gaussian distribution $\mathcal{N}(0, 1)$. One of the prominent properties of PNSs generated in this way is their quasi-orthogonality, i.e.,

$$\mathcal{C}(\mathbf{W}_m, \mathbf{W}_n) \approx \delta(m - n), \tag{5}$$

where $\mathcal{C}(\cdot)$ denotes the operation of linear correlation, which is defined in (4), and $\delta(\cdot)$ is the Delta-function, i.e., $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise.

If each pseudonoise pattern $\mathbf{W}_m$ in the group is used to represent an M-ary message symbol $m \in \{0, \ldots, M - 1\}$, it contains $\log_2 M = N$ bits of information once chosen for data embedding. In other words, the pseudonoise pattern $\mathbf{W}_m$ is modulated by the N bits of data to be embedded. This is the concept of M-ary modulation, also referred to as direct message coding (Cox *et al.*, 2001) and orthogonal modulation (Proakis, 2000) by different authors.

With an additive[2] embedding function, the message m can be embedded into the feature vector $\mathbf{X}$,

$$\tilde{\mathbf{X}} = \mathcal{E}(\mathbf{X}, m) = \mathbf{X} + a\mathbf{W}_m, \tag{6}$$

where $\tilde{\mathbf{X}}$ is the watermarked feature vector, and a is the amplitude factor, controlling the tradeoff between watermark visibility and watermark robustness, which is determined by the requirement of the application.

Now the important issue is how to extract the embedded data from $\tilde{\mathbf{X}}$. If the feature vector $\mathbf{X}$ can be modelled by an i.i.d. sequence with Gaussian distribution, a bank of linear correlators (matched filters) can be applied for optimal extraction of the embedded information, as shown in Fig. 1, where $\mathbf{W}_0, \ldots, \mathbf{W}_{M-1}$ are re-generated PNSs with the same key K as in the embedding process, and $\mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}_i)$, the linear correlation between each reference pattern and the test signal is computed. With a maximum likelihood (ML) estimator, the embedded message is decoded as the index number of the reference pattern which has the maximum correlation with the test signal,

$$\hat{m} = \underset{i\in\{0,\ldots,M-1\}}{\arg\max}\ \mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}_i). \tag{7}$$

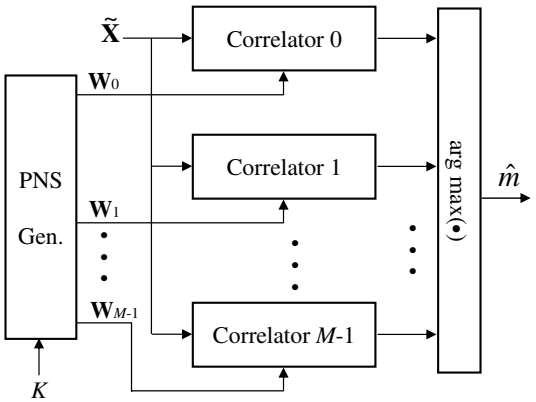


Fig. 1. Structure of the conventional decoder for the extraction of an M-ary watermark.

M-ary message coding can significantly improve the performance of a watermarking system (Kutter, 1999). In general, the greater the value of M, the better the system performance in terms of data error rates or data robustness. However, one issue concerning this decoding method is computation complexity. Because 2^N correlators are needed with an N-bit watermark, the decoder could be computationally prohibitive when N is large. For instance, to extract a 16-bit watermark, 65536 correlations have to be calculated, which could be difficult to implement in practice. Due to this difficulty, a value of $M \geq 256$ appears to be impractical with the decoding structure shown in Fig. 1.

Another M-ary watermark decoding algorithm using a tree-structure was proposed in (Trappe *et al.*, 2003) to reduce the amount of computation. To detect the embedded reference pattern $\mathbf{W}_m$, all the relevant reference patterns are first divided into two $1/2$ size groups

$$\begin{aligned}&\{\mathbf{W}_0, \ldots, \mathbf{W}_{M-1}\}\\ &\quad = \{\mathbf{W}_0, \ldots, \mathbf{W}_{\frac{M}{2}-1}\} \cup \{\mathbf{W}_{\frac{M}{2}}, \ldots, \mathbf{W}_{M-1}\}.\end{aligned} \tag{8}$$

Then the test vector $\tilde{\mathbf{X}}$ is correlated with the sum of all the patterns in each group:

$$\begin{cases} c_1 = \mathcal{C}(\tilde{\mathbf{X}}, \sum\limits_{i=0}^{M/2-1} \mathbf{W}_i) \\ c_2 = \mathcal{C}(\tilde{\mathbf{X}}, \sum\limits_{i=M/2}^{M-1} \mathbf{W}_i). \end{cases} \tag{9}$$

If $c_1 > c_2$, the embedded pattern $\mathbf{W}_m$ must be in the first group, and otherwise in the second group. The group with $\mathbf{W}_m$ is then divided again into two $1/4$ size groups to decide on the location of $\mathbf{W}_m$. This process continues until the exact position of $\mathbf{W}_m$ is located, whose index number is the estimate of the embedded message.

This algorithm reduces the number of correlators to $2\log_2 M$. It should be clear that the actual reduction of computation is less than that, because it introduces some other additional operations, such as summations. An issue related to this approach is that it results in a higher rate of decoding errors than the direct correlation algorithm, especially for blind watermark extraction.

3. M-ary phase modulation for multibit watermarking

As mentioned in the previous section, M correlations for the extraction of an M-ary symbol can be prohibitively expensive when M is large. Another problem inherent in the conventional decoding structure is the time-consuming task of re-generating M independent pseudonoise sequences, $\mathbf{W}_0, \ldots, \mathbf{W}_{M-1}$, which are necessary for data extraction. However, if we drop the requirement on the independence of M pseudonoise sequences, we can solve the problem elegantly with the use of the fast Fourier transform (FFT) and the inverse fast Fourier transform (IFFT), as shown below.

3.1. Multibit watermark via M-ary phase modulation. To overcome the computational bottleneck of the conventional M-ary modulation based watermarking system, we form the set of M reference patterns $\{\mathbf{W}_0, \ldots, \mathbf{W}_{M-1}\}$ with only one reference PNS in the following way:

[2] Another common way to cast a watermark is multiplicative embedding.

- A reference PNS $\mathbf{W}_\mathrm{r}$ is generated as an i.i.d., Gaussian distributed sequence: $W_\mathrm{r}[i] \sim \mathcal{N}(0,1)$, $i = 1,\ldots,L$, where L is the length of the feature vector $\mathbf{X}$.

- Based on $\mathbf{W}_\mathrm{r}$, a set of M PNSs are generated to be circular-shift versions of $\mathbf{W}_\mathrm{r}$, satisfying

$$W_m[i] = \begin{cases} W_\mathrm{r}[i+m] & \text{if } i < L-m, \\ W_\mathrm{r}[i+m-L] & \text{otherwise,} \end{cases} \tag{10}$$

for $m = 0,\ldots,M-1$ and $i = 1,\ldots,L$.

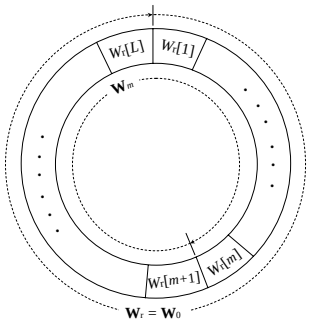


Fig. 2. Formation of a set of circular shift PNSs based on $\mathbf{W}_\mathrm{r}$.

This process is illustrated in Fig. 2. It can be seen that the same PNS can be used to represent M different messages with its M phases, respectively. In other words, a PNS whose phase is modulated by the message m can represent m uniquely. Drawing on the fact that $\mathbf{W}_\mathrm{r}$ is an i.i.d. Gaussian PNS, we can show that the set of PNSs formed in this way satisfy the requirement of quasi-orthogonality expressed by (5), although they are not independent. This property is illustrated by Fig. 3, where, as an example, $\mathbf{W}_\mathrm{r}$ is an i.i.d. normally distributed PNS with 1000 elements, and the correlations of $\mathbf{W}_{200}$ with all the circular-shift versions of $\mathbf{W}_\mathrm{r}$ as a function of the number of shifts are shown.

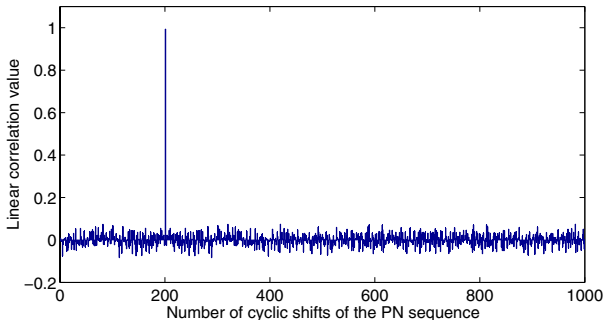


Fig. 3. Linear correlation between a pseudonoise sequence and its circular shift versions.

Now that the set of PNSs $\{\mathbf{W}_0,\ldots,\mathbf{W}_{M-1}\}$ is constructed, we can use its elements for M-ary data hiding according to (6). An interesting part of this proposed algorithm is the extraction of the embedded data. With the circular versions of a PNS as the reference set, we no longer have to perform M correlations separately for data decoding, as is performed conventionally. We can compute, with a very simple method, all the correlations between the watermarked feature vector $\tilde{\mathbf{X}}$ and the M PNSs derived from $\mathbf{W}_\mathrm{r}$. This computation can be implemented conveniently and efficiently by two forward FFT operations and one IFFT operation as follows:

$$\mathbf{c} = \frac{1}{L}\mathcal{F}^{-1}\left(\mathcal{F}(\tilde{\mathbf{X}})\mathcal{F}^{*}(\mathbf{W}_\mathrm{r})\right), \tag{11}$$

where $\mathbf{c} = (c[0], c[1], \ldots, c[L-1])$, $c[i]$ is the correlation between $\tilde{\mathbf{X}}$ and $\mathbf{W}_i$, $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote FFT and IFFT operations, respectively.

The proof of (11) can be found in Appendix. With $c[0],\ldots,c[M-1]$ calculated according to (11), one can immediately get the estimate of the embedded message through (7).

3.2. Multibit watermark via extended *M*-ary phase modulation. It is easy to see that the total number of PNSs derived from a given PNS $\mathbf{W}_\mathrm{r}$ of length L through circular shifting is L. If the desired value of M for M-ary data hiding satisfies $M \leq L$, the efficient method introduced above can be applied. However, if $M > L$, the above scheme does not apply. It appears that at most $\log_2 L$ bits of data can be embedded into the feature vector $\mathbf{X}$ with length L by a pseudonoise sequence. Fortunately this is not true. Next we show that this limitation can be easily circumvented.

Now the set of M reference patterns $\{\mathbf{W}_0,\ldots,\mathbf{W}_{M-1}\}$ are formed in the following way:

- A reference PNS $\mathbf{W}_\mathrm{r}$ is generated as an i.i.d., Gaussian distributed sequence: $W_\mathrm{r}[i] \sim \mathcal{N}(0,1)$, $i = 1,\ldots,M$.

- Based on $\mathbf{W}_\mathrm{r}$, a set of M PNSs are generated to be windowed circular-shift versions of $\mathbf{W}_\mathrm{r}$, satisfying

$$W_m[i] = \begin{cases} W_\mathrm{r}[i+m] & \text{if } i < M-m, \\ W_\mathrm{r}[i+m-M] & \text{otherwise,} \end{cases} \tag{12}$$

for $m = 0,\ldots,M-1$ and $i = 1,\ldots,L$.

This process is illustrated in Fig. 4. It is distinct from the process (10) in two ways. First, the length of $\mathbf{W}_\mathrm{r}$ is M, rather than L. Second, the length of $\mathbf{W}_i, i \in \{0,\ldots,M-1\}$ is less than that of the reference PNS $\mathbf{W}_\mathrm{r}$. In other words, $\{\mathbf{W}_0,\ldots,\mathbf{W}_{M-1}\}$ are derived to be windowed circular shifts of $\mathbf{W}_\mathrm{r}$.

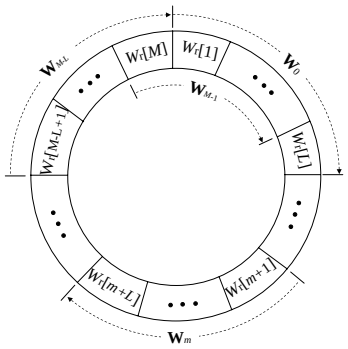


Fig. 4. Formation of a set of windowed circular shift PNSs based on $\mathbf{W}_\mathrm{r}$.

Now let us look at how a multibit watermark is embedded and extracted with the set of PNSs derived by (10). In order to embed an M-ary symbol m, the corresponding $\mathbf{W}_m$ is selected from the set of PNSs, and it is embedded additively into $\mathbf{X}$ according to (6). For watermark extraction, we have to use a slightly different strategy. Since now the watermarked feature vector $\tilde{\mathbf{X}}$ and the reference PNS $\mathbf{W}_\mathrm{r}$ have different lengths, (11) cannot be applied directly. The solution is to first append zeros to $\tilde{\mathbf{X}}$ so that it has the same length as $\mathbf{W}_\mathrm{r}$:

$$\tilde{X}'[i] = \begin{cases} \tilde{X}[i] & \text{for } 1 \le i \le L, \\ 0 & \text{for } L+1 \le i \le M. \end{cases} \tag{13}$$

Then the correlations between $\tilde{\mathbf{X}}$ and the set of PNSs $\{\mathbf{W}_i, i = 0, \ldots, M-1\}$ can be computed by

$$\mathbf{c} = \frac{1}{L}\mathcal{F}^{-1}\left(\mathcal{F}(\tilde{\mathbf{X}}')\mathcal{F}^*(\mathbf{W}_\mathrm{r})\right). \tag{14}$$

Summarizing the solutions to M-ary based data hiding stated above, we give the block diagram of our proposed algorithm for M-ary watermark decoding, which is shown in Fig. 5, where the dashed block means that if $M < L$, the zero-padding process is not necessary, $\otimes$ indicates element-wise product, conj$(\cdot)$ denotes the conjugation operation, and argmax$(\cdot)$ is the function of getting the index number of the largest correlation value.

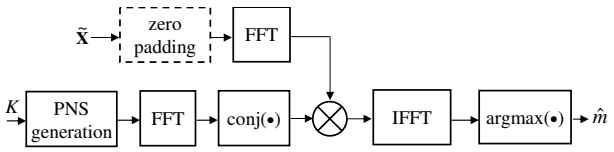


Fig. 5. Structure of the proposed algorithm for efficient M-ary watermark extraction.

3.3. Computational advantage of *M*-ary phase modulation. As noted before, the reason for the adoption of M-ary phase modulation in the design of a watermarking system is that it requires dramatically less computation than a conventional M-ary modulation based system. This computational advantage lies dominantly in the stage of watermark extraction, i.e., data decoding. Now let us compare quantitatively the computational complexity of the proposed method and that of the conventional method. In the case of a conventional M-ary decoder illustrated in Fig. 1, the total number of operations required for the decoding of an M-ary symbol is approximately

$$T_0 = LM, \tag{15}$$

where L is the length of the feature vector. One operation is defined as one real multiplication plus one real addition. Apparently, T_0 is a linear function of M. However, in the case of the proposed M-ary decoder illustrated in Fig. 5, the decoding of an M-ary symbol just involves 2 FFT and 1 IFFT operations. Because the complexity of one FFT or IFFT is $O(M \log_2 M)$ (Jain, 1989), the total number of operations required is approximately

$$T_1 = 3M \log_2 M. \tag{16}$$

To see more clearly the advantage of the proposed M-ary phase modulation over the conventional M-ary modulation, in Fig. 6 we plot T_0 and T_1 as functions of M in the range of our interest, for $L = 1024$.

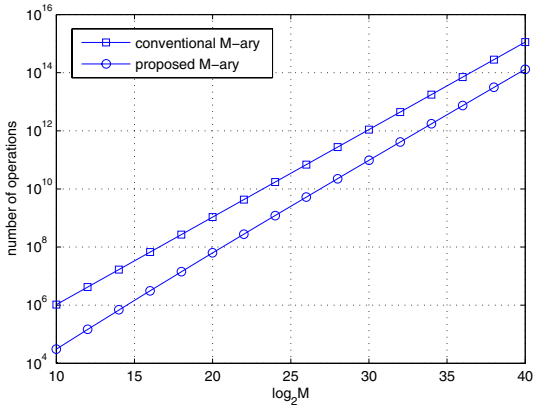


Fig. 6. Algorithm complexity of the conventional M-ary decoder and the proposed M-ary decoder.

One can see from Fig. 6 that algorithm complexity of the conventional M-ary decoder is one or two orders of magnitude higher than that of the proposed M-ary phase decoder when $L = 1024$. On the other hand, T_0 is a linear function of L, but T_1 is independent of L. This means that as L increases, the advantage of the proposed M-ary phase modulation over the conventional M-ary modulation is getting bigger linearly.

If one takes into account the computation involved in the re-generation of PNSs in the conventional M-ary modulation decoder, which is considerable when M is large,

the superiority of the proposed approach of M-ary phase modulation is even more convincing.

4. Performance analysis of *M*-ary watermarks

The algorithm proposed above makes M-ary modulation fully feasible in the design of spread spectrum based watermarking, even if M is very large. Our concern is whether the performance of M-ary data decoding would deteriorate as M increases. We now look into the relationship between the value of M and the error rate of data extraction.

Let $\tilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_m$, where $\mathbf{X}$ is a vector with L i.i.d. elements of $\mathcal{N}(0, \sigma_{\mathbf{X}}^2)$, $\mathbf{W}_m$ is a vector with L i.i.d. elements of $\mathcal{N}(0,1)$, and a is a positive constant. If $\mathbf{W}_k$ is the k-th circular shift of $\mathbf{W}_m$, then it can be shown that

$$\mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}_k) \sim \begin{cases} \mathcal{N}(0, \dfrac{1}{L}(\sigma_{\mathbf{X}}^2 + a^2) & \text{if } k \neq m, \\ \mathcal{N}(a, \dfrac{1}{L}(\sigma_{\mathbf{X}}^2 + 2a^2) & \text{if } k = m, \end{cases} \tag{17}$$

where $\mathcal{C}(\cdot)$ is the correlation function defined in (4).

The proof of (17) can be found in Appendix. Based on this result, we can draw a conclusion about error probability of data extraction.

Let an M-ary message m be embedded into a feature vector $\mathbf{X}$ according to $\tilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_m$, where $\mathbf{X}$ has L i.i.d. elements of $\mathcal{N}(0, \sigma_{\mathbf{X}}^2)$, $\mathbf{W}_m$ is a vector with L i.i.d. elements of $\mathcal{N}(0,1)$, and the constant $a > 0$. If $\sigma_{\mathbf{X}} \gg a$, [3] then the error probability of an ML estimator (7) is

$$P_e \approx 1 - \int_{-\infty}^{\infty} \phi(x) \left[1 - Q(\frac{x}{\sigma_c})\right]^{M-1} \mathrm{d}\, x, \tag{18}$$

where

$$\phi(x) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{(x-1)^2}{2\sigma_c^2}},$$
$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{x^2}{2}}\, \mathrm{d}x,$$
$$\sigma_c = \frac{\sigma_{\mathbf{X}}}{a\sqrt{L}}.$$

The proof of (18) can be found in Appendix. According to (18), we plot the error rate P_e as a function of σ_c^2 for various values of M. In particular, $M = 2^4, 2^8, 2^{12}, 2^{16}$, as shown in Fig. 7. From Eqn. (18) and Fig. 7, we can draw some important conclusions. Firstly, with M and L fixed, P_e is a function of $\sigma_{\mathbf{X}}^2/a^2$, which can be viewed as the signal-to-noise ratio from the perspective of the host signal. It is an intuitive fact that the larger the ratio $\sigma_{\mathbf{X}}^2/a^2$, the weaker the embedded watermark signal, and therefore the more likely the error occurs. Secondly, with M and the ratio $\sigma_{\mathbf{X}}^2/a^2$ fixed, P_e is a function of L. As L increases, the error rate goes down. This is also intuitive, because larger L always reduces the variance of detection statistics, and hence the chance of decoding error. An interesting fact is that L and $\sigma_{\mathbf{X}}^2/a^2$ can be traded with each other. As long as $\sigma_c^2 = \sigma_{\mathbf{X}}^2/a^2 L$ remains unchanged, P_e does not change. Finally, P_e is a function of M. As M increases, the error rate becomes higher. This is a price to pay for the increase in the amount of data embedded.

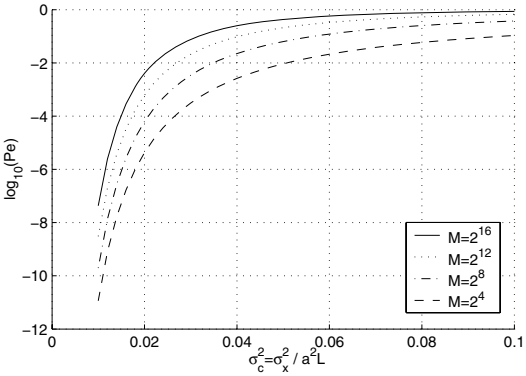


Fig. 7. Error rates of an M-ary ML decoder.

Now we are concerned with the real performance improvement brought by M-ary modulation in our context of watermarking. To have a fair comparison of different cases of M values, we have to fix some parameters, including the number of bits to be embedded N, the power ratio of the feature vector and the watermark $r = \sigma_{\mathbf{X}}^2/a^2$. Under these conditions, there are several schemes to design the watermark, such as FDMA and CDMA approaches, as mentioned in the introduction. Here we focus on the FDMA based M-ary phase modulation approach for the purpose of comparison. The general idea is as follows: An M-ary PNS represents $\log_2 M$ bits of data, and thus for the embedding of N bits into the L-element host vector $\mathbf{X}$, we need to divide $\mathbf{X}$ into $N/\log_2(M)$ subvectors. Each subvector has a length of $L \log_2 M/N$. Different M results in a different number of subvectors, and hence a different length of subvectors. Our goal is to look into the error performance as a function of M. Based on (18), we plot a set of Pe–M curves, fixing $L = 4096, N = 16, r = \{80, 60, 40, 20\}$, as shown in Fig. 8. From this figure, we can see clearly that as M increases, the error rate drops monotonically. This is particularly obvious when r is small, i.e., when the watermark signal is strong.

5. Simulation results

In this section, we apply the proposed M-ary phase modulation technique in the design of a practical watermarking system, from which some experimental results are ob-

[3] This assumption is usually valid due to the requirement of watermark transparency.

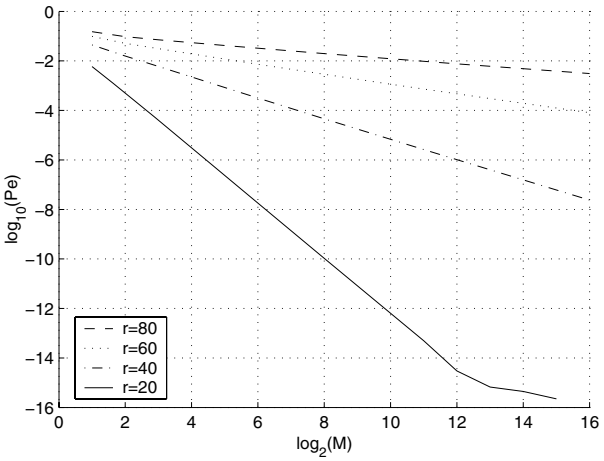


Fig. 8. Error performance of an M-ary watermark vs. the M value.

tained and presented with details. These results verify the effectiveness of the proposed algorithm.

5.1. *M*-ary phase modulation based watermarking system. In order to see the advantage of watermarks based on M-ary phase modulation, we design a multibit watermarking system via a combination of M-ary phase modulation and a CDMA technique. The structure of the watermark embedder is shown in Fig. 9.

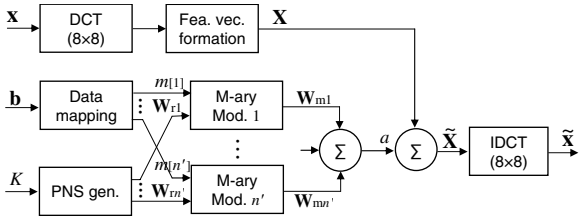


Fig. 9. Embedder structure of the multibit watermarking system based on M-ary phase modulation plus CDMA.

First, an image $\mathbf{x}$ undergoes an 8×8 block DCT transform. In each 8×8 matrix of DCT coefficients, some mid-frequency coefficients are selected for watermarking, as illustrated by Fig. 10. The selected coefficients are subsequently reorganized to be a 1-D feature vector $\mathbf{X}$. A bit sequence $\mathbf{b} = (b_1, \ldots, b_n)$ to be embedded into $\mathbf{X}$ has to be mapped into a sequence of M-ary symbols $\mathbf{m} = (m[1], \ldots, m[n'])$, where $n' = n/\log_2 M$. For each M-ary symbol $m[i]$, a different reference PNS $\mathbf{W}_{\mathrm{r}i}$ is needed, and therefore n' reference PNSs are generated with a key K. The i-th PNS $\mathbf{W}_{\mathrm{r}i}$ is modulated by the M-ary symbol $m[i]$ in the i-th M-ary modulator, in the way described in Section 3, which results in $\mathbf{W}_{\mathrm{m}i}$. Due to the property of quasi-orthogonality, the n' modulated PNSs can be added up based on CDMA. The composite signal

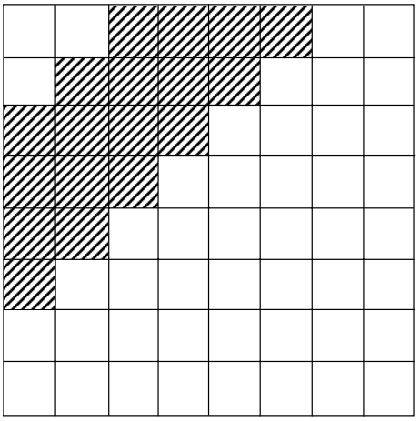


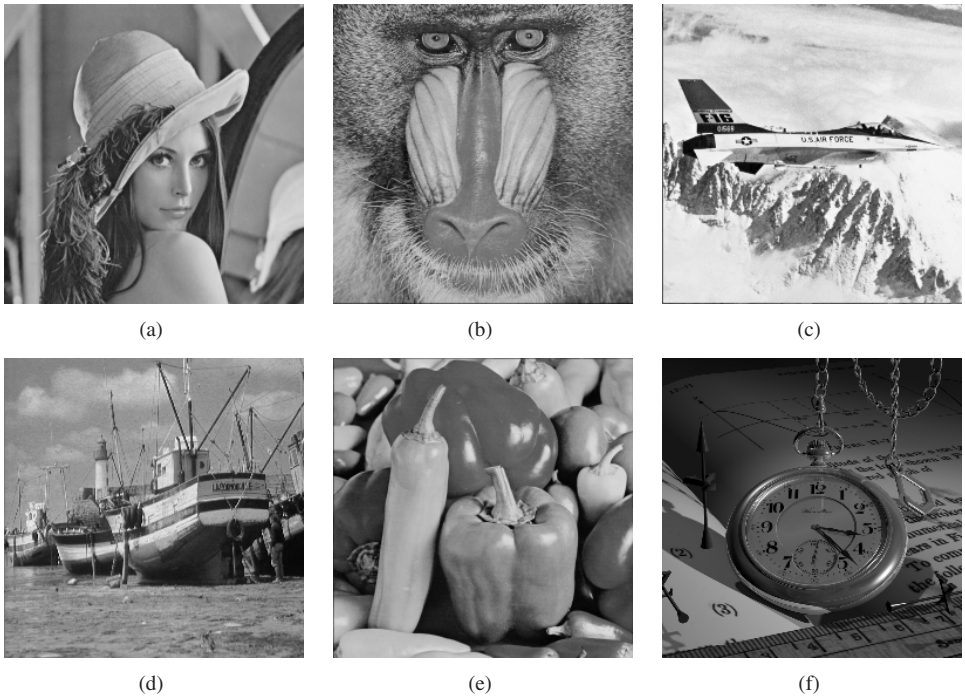
Fig. 10. Coefficients in an 8×8 DCT block selected for data hiding.

$\sum_{i=1}^{n'} \mathbf{W}_{\mathrm{m}i}$ is subsequently scaled by a factor a to control the tradeoff between watermark robustness and watermark obtrusiveness, before it is combined with the feature vector $\mathbf{X}$. Each element in the resulting watermarked vector $\tilde{\mathbf{X}}$ is substituted for its original counterpart in the DCT coefficient matrix, and finally the watermarked image $\tilde{\mathbf{x}}$ is obtained through inverse DCT.

The mechanism shown in Fig. 11 is utilized for watermark extraction. A feature vector $\mathbf{X}'$ is first extracted from a possibly distorted watermarked signal $\mathbf{x}'$ through an 8×8 block DCT transform, and then fed into each of the n' M-ary demodulators. Based on the same key K, the n' reference PNSs are re-generated, and they are used in the n' M-ary demodulators respectively for the estimation of the embedded symbols. The details of each M-ary demodulator are shown in Fig. 5, and explained in Section 3. The estimated M-ary symbols $\hat{m}[i], i = 1, \ldots, n'$, are subsequently mapped into the estimated bit sequence $\hat{\mathbf{b}} = (\hat{b}_1, \ldots, \hat{b}_n)$.

5.2. Experimental results. With the watermark embedder in Fig. 9 and the watermark extractor in Fig. 11, we performed some experiments, focusing on watermark robustness to some common manipulations and the relationship between watermark robustness and the value of M. The test images are a set of 256×256 images with 256 gray levels, shown in Fig. 12. For each experiment in this section, the watermark strength factor a is adjusted such that the quality of the watermarked image remains the same, PSNR = 40dB. The watermark robustness is measured by the bit error rate (BER).

5.2.1. Watermark robustness to lossy compression. Lossy compression of images, dominantly represented by the JPEG standard, is a common and easy way to process images, and therefore watermark robustness against JPEG compression is necessary. An example of JPEG compres-



(a) (b) (c)

(d) (e) (f)

Fig. 12. Original test images: (a) Lena, (b) baboon, (c) F-16, (d) fishing boat, (e) peppers, (f) watch.

(a) (b) (c)

(d) (e) (f)

Fig. 13. Attack examples: (a) JPEG lossy compression, QF=30, (b) cropping, 50%, (c) Gaussian filtering, $5 \times 5, \sigma_g = 1$, (d) Gaussian noise, $\sigma = 10$, (e) salt and pepper noise, $D = 0.05$, (f) histogram equalization.

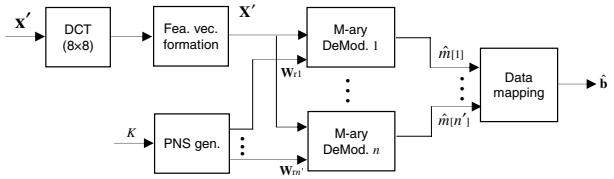


Fig. 11. Decoder structure of the multibit watermarking system based on M-ary phase modulation plus CDMA.

sion is illustrated in Fig. 13(b). To look into the robustness of the designed watermark against JPEG compression, we first watermark images with the data to be embedded, and then compress the watermarked images with a number of different quality factors. The embedded data are estimated by the watermark extraction algorithm possibly with errors from the compressed watermarked images. Another objective of this experiment is to see the relationship between watermark robustness and the value of M. For this purpose, we take $M \in \{2, 4, 16, 256, 65536\}$.

Shown in Fig. 14 are a family of curves of the error performance as a function of JPEG quality factors. Each point on the curves is obtained as the average value of 100 independent experiments, each of which has a different random sequence of 64 bits as its data input. From Fig. 14 one can see that with the increase in quality factor, the BER drops monotonically. An important trend is that the value of M influences the BER significantly. In particular, larger M gives a lower BER. This result evidently shows that M-ary modulation is preferable in the design of a multibit spread spectrum-based watermarking system.

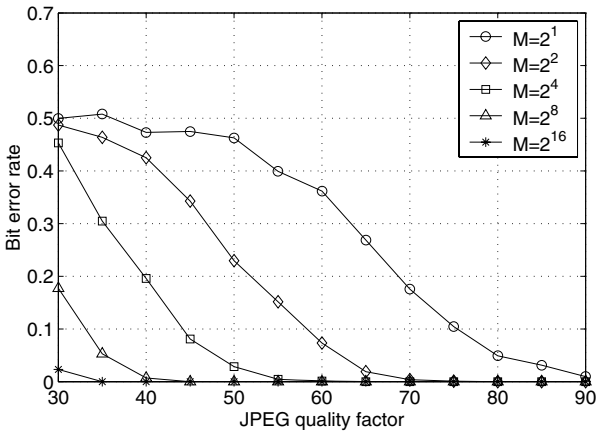


Fig. 14. Error performance of the multibit watermarking system based on M-ary modulation plus CDMA, under JPEG lossy compression. The number of bits embedded is 64, and the quality of watermarked images is PSNR = 40dB.

5.2.2. Watermark robustness to image cropping. Image cropping refers to the loss of some parts of an image, especially along the borders. An example of image cropping is illustrated in Fig. 13(b). Image cropping brings about a partial loss of watermark information. The objective of this experiment is to look at the system's ability to recover the embedded data from incomplete watermarked images. Preferably the embedded data can be extracted at a low error rate under mild image cropping. In our experiments, we crop the watermarked images evenly along the four borders to different degrees, and record the errors in data extraction from the cropped images. The amount of data embedded is 128 bits. Shown in Fig. 15 are a family of BER curves, with $M \in \{2^4, 2^8, 2^{16}\}$, as a function of the remaining factor, which is the ratio of the number of remaining pixels to that of original pixels. From the figure, one can see that the watermark has outstanding robustness to image cropping, especially when $M = 2^{16}$. Even if 75% of the image pixels are cropped, the embedded data can still be extracted with a very low BER at the magnitude of $O(10^{-3})$.

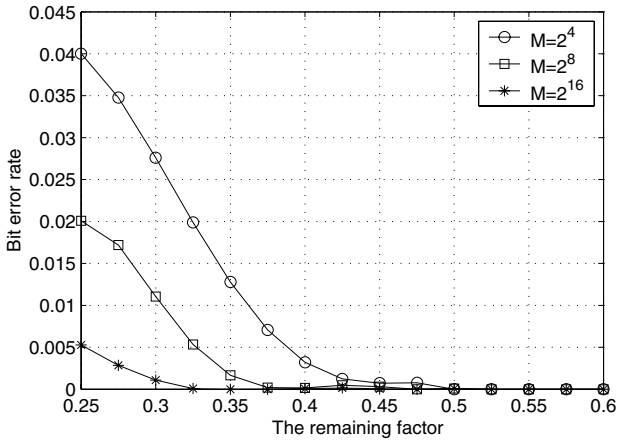
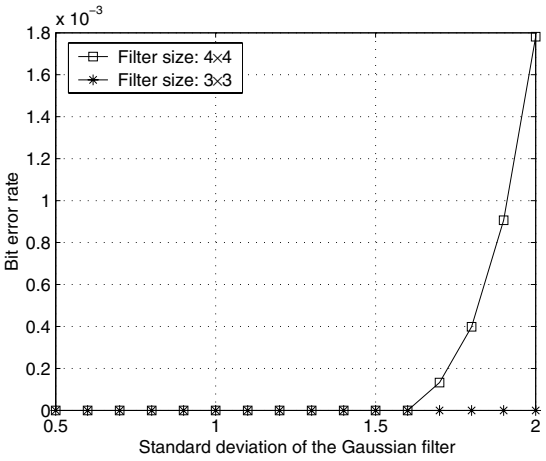


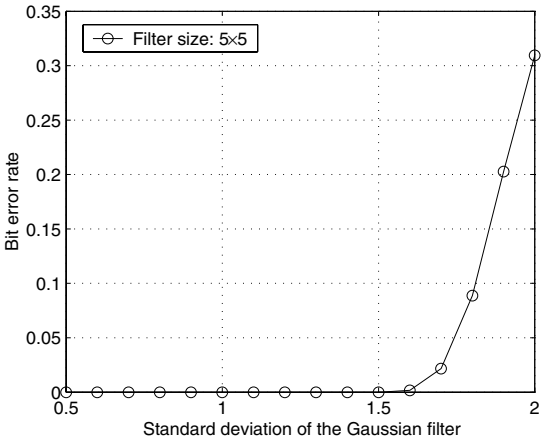
Fig. 15. Error performance of the watermark under image cropping. The number of bits embedded is 128, and the quality of watermarked images is PSNR = 40dB.

5.2.3. Watermark robustness to lowpass filtering. Lowpass filtering is another common form of image processing, which can be performed conveniently either in a transform domain or directly in a space domain (Gonzalez and Woods, 2002). Here we use a Gaussian filter to test watermark robustness to this kind of attack against watermarked images. One such attack example is illustrated in Fig. 13(c). We apply 2^{16}-ary phase modulation, set the length of data to be 128 bits and PSNR=40dB in all the experiments. Figure 16(a) shows the test results in the cases of 3×3 and 4×4 filter sizes, while the results for 5×5 Gaussian filters are given in Fig. 16(b). The standard deviation of the Gaussian filter is chosen to cover a wide range: $0.5 < \sigma_{\mathrm{g}} < 2$. The results are the average of 1000 repetitions. In all our experiments, BER=0 in the case of 3×3 filters regardless of σ_{g}, and BER=0 if $\sigma_{\mathrm{g}} \leq 1.5$ in

the cases of 4×4 and 5×5 filters. These results indicate that the designed watermark has outstanding robustness against the attack of lowpass filtering.



(a)



(b)

Fig. 16. Error rate as a function of the standard deviation of the Gaussian filter: (a) $\sigma_g = 3$ and 4, (b) $\sigma_g = 5$.

5.2.4. Watermark robustness to other attacks. Besides the attacks considered above, we are concerned about watermark robustness to some other kinds of attacks as well. A set of common image manipulations, including noise addition and image enhancement operations illustrated in Fig. 13, are applied to the watermarked images in order to test watermark robustness. Table 1 lists the error rates under these attacks. Throughout all the tests, we use 2^{16}-ary phase modulation, embed 128 bits of data and make PSNR=40dB. The table shows the embedded data are robust enough against most commonly used image processing operations.

Table 1. Watermark robustness to other common attacks.

Type of attack	Parameter of attack	BER
White Gaussian noise	$\sigma = 5$	0
	$\sigma = 10$	0
	$\sigma = 15$	0
Salt & pepper noise	$D = 0.01$	0
	$D = 0.03$	0
	$D = 0.05$	3.33×10^{-2}
Histogram equalization	N/A	0
Median filtering	f. size $= 2 \times 2$	0
	f. size $= 3 \times 3$	0
	f. size $= 4 \times 4$	5.89×10^{-2}
Wiener filtering	f. size $= 2 \times 2$	0
	f. size $= 3 \times 3$	0
	f. size $= 4 \times 4$	7.19×10^{-4}
Sharpening (in Paintshop Pro)	Moderate	0
	High	0

6. Conclusions

In this paper, we proposed to design a multibit watermarking system based on M-ary phase modulation. The conventional use of M-ary modulation has been limited by small M values, e.g., $M \leq 256$, due to heavy computations associated with correlation based signal detection. However, with the proposed M-ary phase modulation, which is based on circular shifts of a reference PNS, the amount of computation in watermark detection is drastically reduced. Furthermore, we also provided the design of an extended M-ary phase modulated watermark based on a set of windowed circular shifts of a PNS of length M, which breaks the restriction on the value of M due to the length of the feature vector. A practical design of a multibit watermark based on M-ary phase modulation plus CDMA was presented. The simulation results showed that the proposed M-ary phase modulation greatly improved the tradeoff among a watermark's transparency, robustness and information capacity while keeping a low cost of implementation.

References

Cheng Q. and Huang T. S. (2001). An additive approach to transform-domain information hiding and optimum detection structure, *IEEE Transactions on Multimedia*, **3**(3): 273–284.

Cox I. J., Killian J., Leighton T. and Shanmoon T. (1997). Secure spread spectrum watermarking for multimedia, *IEEE Transactions on Image Processing*, **6**(12): 1673–1687.

Cox I. J., Miller M. L. and Bloom J. A. (2001). *Digital Watermarking*, Morgan Kaufmann, San Francisco.

Gonzalez R. and Woods R. (2002). *Digital Image Processing*, Prentice-Hall, New York.

Hernandez J. R., Amado M. and Perez-Gonzalez F. (2000). DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure, *IEEE Transactions on Image Processing*, **9** (1): 55–68.

Jain A. K. (1989). *Fundamentals of Digital Image Processing*, Prentice-Hall: Englewood Cliffs, NJ.

Kutter M. (1999). Performance improvement of spread spectrum based image watermarking schemes through M-ary modulation, *Lecture Notes in Computer Science*, **1728**: 238–250.

Nikolaidis A. and Pitas I. (2003). Asymptotically optimal detection for additive watermarking in the DCT and DWT domains, *IEEE Transactions on Image Processing*, **12**(5): 563–571.

O'Ruanaidh J. and Pun T. (1998). Rotation, scale and translation invariant spread spectrum digital image watermarking, *Signal Processing*, **66**(8): 303–317.

Proakis J. G. (2000). *Digital Communications, 4th Ed*, McGraw Hill: New York.

Trappe W., Wu M., Wang Z. J. and Liu K. J. R. (2003). Anti-collusion fingerprinting for multimedia, *IEEE Transactions on Signal Processing*, **51**(4): 1069–1087.

Wilson S. G. (1996). *Digital Modulation and Coding*, Prentice Hall, New York.

Zeng W. and Liu B. (1999). A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images, *IEEE Transactions on Image Processing*, **8**(11): 1534–1548.

Appendix

A.1 Derivation of Eqn. (11) The linear correlation between $\mathbf{X}$ and $\mathbf{W}_k$ is

$$c[k] = \frac{1}{L}\sum_{i=0}^{L-1} X[i]W_k[i] = \frac{1}{L}\sum_{i=0}^{L-1} X[i]W_0[i-k], \quad k = 0, \ldots, L-1. \tag{19}$$

Its DFT is

$$\begin{aligned} C[u] =& \frac{1}{L}\sum_{k=0}^{L-1}\left[\sum_{i=0}^{L-1} X[i]W_0[i-k]\right] e^{-j\frac{2\pi}{L}uk} \\ =& \frac{1}{L}\sum_{i=0}^{L-1} X[i]\sum_{k=0}^{L-1} W_0[i-k]e^{-j\frac{2\pi}{L}uk} \\ =& \frac{1}{L}\sum_{i=0}^{L-1} X[i]e^{-j\frac{2\pi}{L}ui}\sum_{k=0}^{L-1} W_0[i-k]e^{j\frac{2\pi}{L}u[i-k]} \\ =& \frac{1}{L}\mathcal{F}(\mathbf{X})\mathcal{F}^*(\mathbf{W}_0), \quad u = 0, \ldots, L-1, \end{aligned} \tag{20}$$

which leads to

$$c[k] = \frac{1}{L}\mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{X})\mathcal{F}^*(\mathbf{W}_0)\right), \quad k = 0, \ldots, L-1. \tag{21}$$

In the above equations, $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote DFT and IDFT operations, respectively.

A.2 Derivation of Eqn. (17) The correlation between $\tilde{\mathbf{X}}$ and $\mathbf{W}_k$ is

$$\mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}_k) = \mathcal{C}(\mathbf{X}, \mathbf{W}_k) + a\,\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k). \tag{22}$$

Let us look at the first term on the right-hand side. According to the Central Limit Theorem, $\mathcal{C}(\mathbf{X}, \mathbf{W}_k)$ follows a Gaussian distribution when L is sufficiently large. Based on the fact that $\mathbf{X}$ and $\mathbf{W}_r$ are independent, the mean and variance of $\mathcal{C}(\mathbf{X}, \mathbf{W}_k)$ can be obtained:

$$E\{\mathcal{C}(\mathbf{X}, \mathbf{W}_k)\} = E\Big\{\frac{1}{L}\sum_{i=0}^{L-1} X[k]W_k[i]\Big\} \tag{23}$$

$$= \frac{1}{L}\sum_{i=0}^{L-1} E\Big\{X[i]\}E\{W_k[i]\Big\} = 0. \tag{24}$$

$$V\Big\{\mathcal{C}(\mathbf{X}, \mathbf{W}_k)\Big\} = E\left\{\left(\frac{1}{L}\sum_{i=0}^{L-1} X[i]W_k[i]\right)^2\right\} \tag{25}$$

$$= \frac{1}{L^2}\sum_{i=0}^{L-1} E\{(X[i])^2(W_k[i])^2\} + \frac{1}{L^2}\sum_{\{(i,j),i\neq j\}} \underbrace{E\{(X[i]W_k[i])(X[j]W_k[j])\}}_{=0} \tag{26}$$

$$= \frac{1}{L^2}\sum_{i=0}^{L-1} \underbrace{E\{(X[i])^2\}}_{=\sigma_{\mathbf{X}}^2}\underbrace{E\{(W_k[i])^2\}}_{=1} = \frac{\sigma_{\mathbf{X}}^2}{L}. \tag{27}$$

We can analyze the second term on the right-hand side of (22) in a similar way. When $k \neq m$, we have

$$E\{\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k)\} = 0, \tag{28}$$

$$V\{\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k)\} = \frac{1}{L}. \tag{29}$$

When $k = m$, we get

$$E\{\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k)\} = \frac{1}{L}\sum_{i=0}^{L-1} E\{(W_m[i])^2\} = 1, \tag{30}$$

$$V\{\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k)\} = E\left\{\frac{1}{L^2}\left(\sum_{i=0}^{L-1}(W_m[i])^2\right)^2\right\} - 1 \tag{31}$$

$$= \frac{1}{L^2}\sum_{i=0}^{L-1} E\{(W_m[i])^4\} + \frac{2}{L^2}\sum_{\{(i,j),i\neq j\}} \underbrace{E\{(W_m[i])^2\}E\{(W_m[j])^2\}}_{=1} - 1 \tag{32}$$

$$= \frac{1}{L^2}3 + \frac{2}{L^2}\frac{L(L-1)}{2} - 1 = \frac{2}{L}. \tag{33}$$

Note that in (32),

$$E\{(W_m[i])^4\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} x^4 e^{-x^2/2}\,\mathrm{d}x = 3,$$

and there are $\binom{L}{2}$ products in total involved in the second summation.

Combining (22), (24), (27)–(30) and (33), we obtain

$$\mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}_k) \sim \begin{cases} \mathcal{N}\Big(0, \dfrac{1}{L}(\sigma_{\mathbf{X}}^2 + a^2 L)\Big) & \text{if } k \neq m, \\ \mathcal{N}\Big(a, \dfrac{1}{L}(\sigma_{\mathbf{X}}^2 + 2a^2)\Big) & \text{if } k = m. \end{cases} \tag{34}$$

A.3 Derivation of Eqn. (18) Let $\mathbf{c} = (c[0], \ldots, c[M-1])$ where $c[k] = \mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}_k)$. According to Theorem 17, we have

$$\frac{c[m]}{a} \sim \mathcal{N}(1, \sigma_1^2),$$
$$\frac{c[i]}{a} \sim \mathcal{N}(0, \sigma_0^2), \quad i \in \{0, \ldots, M-1\} \text{ but } i \neq m,$$

where

$$\sigma_1^2 = \frac{r+2}{L}, \quad \sigma_0^2 = \frac{r+1}{L}, \quad r = \frac{\sigma_{\mathbf{X}}^2}{a^2}.$$

When $\sigma_{\mathbf{X}}^2 \gg a^2$, we have $\sigma_1^2 \approx \sigma_0^2 \approx r/L$.

Let

$$c_{\max} = \max_{\{i, i \neq m\}} \{c[i]\}.$$

$F_{\max}(x)$ and $F_i(x)$ denote the distribution functions of $c_{\max}/a$ and $c[i]/a$, respectively. Then

$$F_{\max}(x) = \prod_{\{i, i \neq m\}} F_i(x).$$

Thus

$$P\Big(\frac{c_{\max}}{a} < x\Big) = \prod_{i=0, \neq m}^{M-1} P\Big(\frac{c[i]}{a} < x\Big) = \Big[1 - Q(\frac{x}{\sigma_c})\Big]^{M-1},$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-\frac{x^2}{2}}\,\mathrm{d}x, \quad \sigma_c = \sqrt{\frac{r}{L}} = \frac{\sigma_{\mathbf{X}}}{a\sqrt{L}}.$$

Following the ML principle in (7), the embedded symbol m is correctly decoded only when $c[m] > c_{\max}$, and therefore the probability of correct estimation is

$$P_c = \int_{-\infty}^{\infty} \phi(x) P\Big(\frac{c_{\max}}{a} < x\Big)\,\mathrm{d}x,$$

where

$$\phi(x) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{(x-1)^2}{2\sigma_c^2}}$$

is the probability density function of $c[m]/a$. Finally, we have

$$P_e = 1 - \int_{-\infty}^{\infty} \phi(x) \Big[1 - Q(\frac{x}{\sigma_c})\Big]^{M-1}\,\mathrm{d}x.$$