International Journal of Computer Science
and Engineering (IJCSE)
ISSN 2278-9960
Vol. 2, Issue 1, Feb 2013, 29-46

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

ROBUST AUDIO WATERMARKING USING DISCRETE WAVELET AND DISCRETE COSINE TRANSFORMS

HOOMAN NIKMEHR[1] & SINA TAYEFEH HASHEMY[2]

[1]Department of Computer Engineering, University of Isfahan, Isfahan, Iran

[2]Department of Computer Engineering, Bu-Ali Sina University, Hamedan, Iran

ABSTRACT

Audio watermarking hides copyright information into the digital audio signal. Embedded data not only must be imperceptible but also should resist attacks and other types of distortions trying to remove or neutralize the watermark picture. this paper presents a novel audio watermarking scheme. This algorithm, divides the digital audio signal into 2-section segments. Then, for each segment, the first section passes through a DWT block and meanwhile the second is processed by a DWT followed by a DCT block. While the algorithm hides the synchronization bits into the DWT coefficients of the first section it embeds the watermark bits into the DCT coefficients. Unlike other methods, the proposed approach uses a 4-point quantization technique for embedding the bits. Simulations show that while the noise introduced by the new scheme to the audio signal is imperceptible, the watermarked audio signal is more robust against attacks compared to other methods.

KEYWORDS: DCT, DWT, IMPERCEPTIBILITY, ROBUSTNESS, SNR

INTRODUCTION

Due to the fast improvement in the communication technology, accessing (including downloading, copying, sharing and exchanging) digital audio files over the Internet is becoming easier everyday. Having considered these circumstances, protecting the intellectual property rights of such digital works has become a hot research topic that requires urgent solution. One of the most promising approaches that has seriously attracted the researchers in the recent years is digital watermarking [1].

Digital watermarking is a technique, which hides digital copyright information into the digital media so that the inserted bits are not recognizable by human senses. It is essential that the embedding mechanism is so robust that can resist common intentional or unintentional attacks. A considerable research focusing on image and video watermarking has been carried out, however, only a few algorithms have been reported on audio watermarking since human's auditory system (HAS) is more sensitive than human's visual [1].

Almost all audio watermarking algorithms take advantage of the weakness of HAS or the perceptual properties of HAS to embed the watermark data in the parts of audio signals so that the distortion resulted from the watermarking process is not audible. These techniques can be categorized into time domain techniques such as

- hiding watermark information in low order bits of the audio [2],
- echo hiding [3],
- spread spectrum [4], [5], and
- patchwork algorithms [6], [7],

watermarking by modifying the audio's frequency coefficients like

- phase coding
- coefficient quantization, and embedding the watermark into the audio signal by modifying the coefficients of the transforms' outputs such as
- Short-Time Fourier Transform (STFT) [8],
- Discrete Wavelet Transform (DWT) [9]–[11] and
- Discrete Cosine Transform (DCT) [12].

Time domain watermarking has a higher payload (number of bits hidden in a slice of audio signal represented commonly as bps) compared to watermarking schemes of other domains, however, since the embedding process performed directly on the original host audio signal (or selected blocks of the audio signal) requires a great number of threshold values to be defined [13], time domain methods are vulnerable against attacks like resampling, low pass filtering and compressing. This means that the hidden watermark can be easily destroyed since attacks based on common signal processing procedures can easily change the threshold values that are required for correct extracting the watermark from a watermarked audio signal.

In the spread spectrum method, the digital watermark bits are modulated using a pseudo-random sequence and then the new signal is added to the original audio signal [2]. In [4], a number of innovative audio watermarking methods based on the spread spectrum algorithm are introduced. The proposed methods adjust the modulated watermark bits with the frequency response of HAS so that the HAS shaped signal is kept in the frequency range that HAS is the least sensitive to. Then, the resulted signal is inserted into the audio using MCLT (Modulated Complex Lapped Transfer) function. Although, this method is one of the most popular audio watermarking algorithms that has been studied in detail in the literature [14]–[17], its need to a large number of threshold values makes it hard to be practical in real applications [13]. In addition, as the main disadvantage, spread spectrum scheme requires having access to the host audio signal to extract watermark information efficiently.

In the phase coding approach [2], [18]–[20] the audio signal is equally split into segments. Then, the first segment's phase coefficients are replaced with those of the reference phase that represent the digital watermark bit sequence. Then, the phase coefficients of the subsequent segments are adjusted so that the phase relationship is preserved. This makes the induced phase distortion minimally invasive. However, the phase coding audio watermarking method is not robust against attacks manipulating the phase. In addition, since this method does not distribute the watermark all over the entire host audio signal, cropping attacks can easily eliminate the watermark information.

Although, the transforms have been commonly employed in digital image watermarking for a long time, they have been just recently introduced to digital audio watermarking [21]–[24]. Among them, there is DWT, which is very popular in digital signal processing (DSP) applications. In audio watermarking using DWT, the audio signal is decomposed into several frequency sub-bands. Then the digital watermark bits are inserted into the coefficients of one or more sub-bands that are obtained in the previous step using the quantization techniques. The quantization process quantizes a coefficient based on a quantization step (threshold) that corresponds to arbitrary binary value 0 or 1.

The decomposition process can be performed in different levels. The more the levels are, the more accurate DWT in terms of the frequency and the time is obtained. In other words, since in higher levels of DWT, the frequency ranges of the sub-band become narrower, signal can be represented in detail with more precise frequencies. DWT is an O(n) process, which is faster than of the other transforms [25] and also it is reasonably robust against compressing attacks such as mp3, however, since DWT directly modifies the transform coefficients, some distinguishable artifacts can be induced in the low energy regions of the host audio signal.

DCT is another transform that has been widely used recently in signal compressing algorithms because of its considerable capability in compressing the signal energy even in few coefficients [26]. In audio watermarking applications, DCT embeds the digital watermark into the coefficients obtained from the transform using the quantization techniques. This approach has a high SNR since the digital watermark bits are inserted in the high energy sections of the host audio signal resulting in a very clear watermarked sound. Moreover, DCT is robust against resampling and low pass filtering attacks. However, it is vulnerable to compressing attacks such as mp3 since in the heart of every audio compressing algorithm an energy compressing transform such as DCT can be found. To improve the performance and the robustness of the audio watermarking technology it is possible to combine two or more of the previously described methods. Wang and Zhao [27] propose an approach, which employs DWT and DCT together for audio watermarking. The combined technique shows high SNR. This means that the quality of the watermarked audio is well preserved and the hidden watermark bits are inaudible. In addition, this technique is robust against the compressing attack like mp3, however, there are a number of disadvantages as follows.

- The information required for synchronization during the watermark extraction process is embedded into the host signal in the time domain, which is weak against most attacks such as resampling, filtering and compression. This can lead to an unsuccessful synchronization process since in the extraction phase, the algorithm is not able to find the exactly position of the segment in where the digital watermark bits are inserted.
- DCT process is applied on a short range of the host audio signal. This means that there is chance that watermark information is embedded into the low energy parts of the host audio signal. This can make the watermark weak against the filtering attacks.

The motivation of this research is based on the approach presented in [27]. DCT and DWT are used side by side to implement an audio watermark algorithm, however, unlike [27], to improve the synchronization stability and the watermark robustness, instead of inserting the synchronization code into the host audio in the time domain, it is embedded in the wavelet coefficients of the host audio. The main features of the new method are as follows.

- In the first step, DWT is applied to the host audio.
- Then, the synchronization code bits are embedded into the DWT coefficients.
- Afterwards, DCT is used to compress the energy of the coefficients obtained in the first step.
- Finally, the digital watermark bits are inserted into the DCT coefficients using the quantization technique.
- In addition, to increase SNR and to improve the stability of the new method, watermark is inserted into segments of the audio that are lengthier than those are commonly used.
- Moreover, the presented approach employs DCT to insert watermark information in high energy coefficients.
- This technique takes advantage of a 4-point quantization [9], which gives more satisfying results than those of a 2-point quantization.

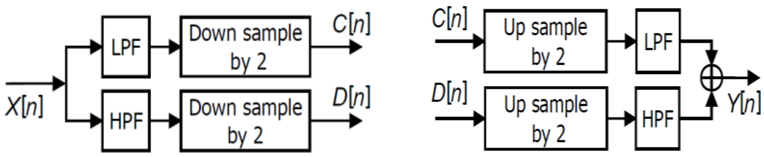
This paper is organized as follows. In Section 2, DWT method is briefly described. Section 3 is spent on DCT.

Sections 4 and 5 discuss some requirements used for embedding/extracting the watermark into/from the audio signal. The new audio watermarking method is proposed in Section 6 Evaluation metrics are described in Section 7 while comparisons between results of evaluating the proposed audio watermarking scheme and those of the previously published methods are shown and discussed in Section 8 Finally, Section 9 concludes the paper.

DWT

DWT is utilized in a wide range of DSP applications including audio/image/video compression, data communication over the Internet, pattern recognition and numerical analysis [25]. This transform can effectively represent signals especially those have localized variations. For example, consider representation of the unit impulse function using the Fourier transform. This representation requires an infinite number of sinusoidal terms due to the very rapid change in the signal. However, DWT can demonstrate the unit impulse signal using only a few terms.

In DWT, each level is called an octave, which at least in a 1-D case, can be constructed as a pair of finite impulse response (FIR) filters; a low pass filter (LPF) and a high pass filter (HPF), as shown in Figure 1(a). Inside an octave, two down-sampling blocks operate, each after a filter. These two halve the output samples of the octave and resulting in a minimized calculation load. In contrast, the reverse of DWT can be formed as an inverse LPF (ILPF) and an inverse HPF (IHPF) following two up-sampling blocks as depicted in Figure 1(b).



(a) DWT b) IDWT

Figure 1: Wavelet Transform Blocks

In the DWT diagram, while the LPF produces the mean of the signal, the HPF extracts the signal's details. To make it clear, suppose that the LPF in Figure 1(a) has coefficients of $\{0.5, 0.5\}$ .This means that the filter's function is equal to

$$\frac{x[n]+x[n-1]}{2}, \tag{1}$$

which is exactly the mean of the values of two successive samples. Correspondingly, if the HPF in Fig 1(a) is assumed to have coefficients of $\{0.5, -0.5\}$ the output of the HPF is

$$\frac{x[n]-x[n-1]}{2}, \tag{2}$$

which is equal to the half of the difference between the values of two consecutive samples. This means that while the mean signal is very similar to the original signal, the detailed signal, which is produced by the HPF, is also required in order to reconstruct the original signal. In multiresolution analysis of the signals, the mean signal from the first octave is applied to a pair of LPF/HPF filters in the second octave. This results in a new pair of mean/detailed signals in the second octave. However, from this point onwards, in every octave (except the last one), only the detailed signals are kept and the mean signals are discarded. Therefore, the number of output samples of a wavelet octave, is half the number of input mean signal. It should be noted that the input mean signal to an octave can be reconstructed using the output mean/detailed signals [25]. This implies that to reproduce the original audio signal, DWT only needs to keep all detailed signals along with the last output mean signal.

DWT shows advantages of both short-time Fourier transform (STFT) and discrete-time Fourier transform (DTFT) at the same time [25]. Compared to the transforms analyzing signals in the time-frequency domain, DWT is more accurate in displaying high frequencies.

DCT

DCT is a transform representing a signal in the form of a series of coefficients obtained from a sum of cosine functions oscillating at different frequencies and at different amplitudes [26]. Like the other transforms, DCT is applied to remove the correlation among the elements of the signal. For a 1-D discrete signal with the length of N, DCT is expressed as

$$C(m) = a(m) \sum_{n=0}^{N-1} f(n) \cos\left[\frac{\pi(2n+1)m}{2N}\right], \tag{3}$$

where m = 0, 1, 2,…, N-1. For the 1-D signal, reverse DCT is represented as

$$f(n) = \sum_{m=0}^{N-1} a(m)C(m) \cos\left[\frac{\pi(2m+1)n}{2N}\right] \tag{4}$$

In (3) and (4), for m = 0, 1, 2,…, N-1, $a(m)$ is

$$a(m) = \begin{cases} \sqrt{\dfrac{1}{N}}, & \text{if} \quad m = 0 \\ \sqrt{\dfrac{2}{N}}, & \text{if} \quad m \neq 0 \end{cases} \tag{5}$$

Ability in compressing energy of the signal, in few coefficients, is one of the criteria for comparing performance of the transforms. DCT is among the best and therefore, when quantizing, the transform is allowed to ignore the coefficients with low amplitudes without losing the accuracy during reconstructing the signal from its coefficients.

SYNCHRONIZATION

In most audio watermarking algorithms, the digital watermark bits are embedded into specific positions of the host audio signal. Therefore, to detect the hidden bits, the extracting process needs to know their positions. This is called the synchronization problem [13]. Synchronization is the key issue during watermark extraction process especially when the host audio is manipulated by desynchronizing attacks [13]. Any shift in the bits positions makes extracting schemes unable to succeed.

The main goal of the synchronization schemes is to find where the new shifted positions are. In the first step of a watermark extraction procedure, the detecting process tries to align itself with the watermarked block. If it fails, it is impossible to extract the watermark bits from the host audio, causing a false detection. In practice, time or frequency scaling attacks can lead to desynchronization. Therefore, to have a robust audio watermarking, it is required to employ a synchronization technique that can resist such attacks. A very common technique inserts synchronization codes into the audio signal [7], [9], [14], [27].

When choosing a synchronization code, the following three issues need to be considered,

- How the code bits are inserted into the audio signal.

- The number of code bits.
- Distribution of 0s and 1s over the code length.

Among these three, the length of the synchronization code has a direct relationship with improving its robustness [27].

PREPROCESSING

In most practical audio watermarking techniques, the watermark is first preprocessed using Arnold transform [6], [27]–[30]. This increases the security of the embedded data by scrambling the 2-D structure of the digital picture. The rest of the preprocessing is as follows.

- To improve the robustness of the audio watermarking especially against the cropping attack, even if the synchronization is lost, the original audio signal is first divided into equal segments called A^i.
- Segment A^i is divided into sections A_1^i and A_2^i with lengths of L_1 and L_2 respectively, as shown in Figure 2. In this step, the synchronization code bits are inserted into A_1^i while the digital watermark bits are hidden inside A_2^i.

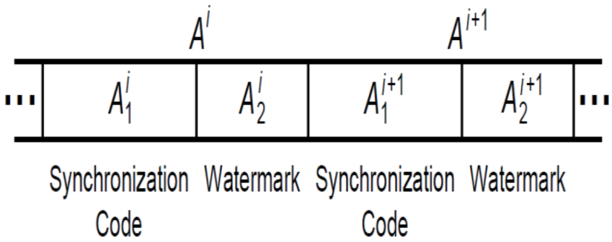


Figure 2: Signal Segmentation

- The above is repeated for every audio segment A^i, therefore if a attack corrupts few audio segments, the watermark still can be extracted through the remaining segments.

PROPOSED AUDIO WATERMARKING METHOD

As illustrated in Figure 3, the proposed audio watermarking approach takes advantages of both DWT and DCT in two phases as follows.

1. Using DWT algorithm, segments of the host audio are selected so that they are in a frequency range that when they are manipulated (by embedding digital watermark bits) the resulted distortion is not audible. DWT uses the HAS specification to determine the appropriate frequency range.
2. The output signal of the first phase is applied to a DCT to calculate the DCT coefficients. However, only high energy coefficients are chosen for watermark insertion. This is to make sure that none of the low energy segments of the host audio is affected by the embedding process, and therefore, to guarantee a high quality less distorted watermarked audio.
3. Like [27], the watermark bits are embedded into the host audio in three steps as follows.

- The audio signal is divided into segments, each segment into two sections.
- DWT is applied to the first section of each segment and the coefficients are calculated. Then the synchronization bits are inserted into the DWT coefficients using the coefficient quantization technique.

- DWT and then DCT are applied on the second section of each segment. This is followed by the coefficient quantization, which results in embedding the digital watermark bits into the low frequency DCT coefficients.

The proposed approach uses a Barker code [31] as the synchronization code to find the position of the digital watermark bits hidden in the host audio signal. These codes are commonly used in the synchronization of data frames in digital communication systems. Side-lobe correlation of Barker codes is low. This means that a Barker code and a time shifted (not aligned) copy the code are not very correlated (the code's autocorrelation is not grater than 1). Although [27] uses a 16-bit Barker code, it is proven that there is no Barker code longer than 13 bits.

Algorithm

Let A represent a digital audio signal as

$$A = \{a(n), 1 \le n \le N\}, \tag{6}$$

where N is the number of signal samples. The watermark (a digital picture) can be shown as

$$W = \{w(i, j), 1 \le i \le P, 1 \le j \le Q\}, \tag{7}$$

where P and Q represent the picture's width and length and $w(i, j) \in \{0,1\}$ is the value of a pixel in the Cartesian plane with i and j axes. Similarly, the synchronization code F is defined as

$$F = \{f(t), 1 \le t \le K\}, \tag{8}$$

where K is the number of code's bits. The proposed audio watermarking algorithm comprises the phases described in the following subsections.

Synchronization Code Insertion

In this section, the synchronization code insertion method used in the proposed audio watermarking algorithm is described. First, H-level DWT is applied to every section A_1^i in order to obtain the DWT coefficients as

$$A_1^{iH}, D_1^{iH}, D_1^{i(H-1)}, \ldots, D_1^{i1} \tag{9}$$

where D_1^{ij} represents the detailed signal resulted from the j-th DWT level and term A_1^{iH} is an approximation of signal A_1^i and is where the synchronization code is inserted in. In the proposed audio watermarking method, for inserting the synchronization code, a 4-point quantization is used as

$$a(t)_1^{i'H} = \begin{cases} a(t)_1^{iH} - R_1 + \frac{S_1}{4}, & \text{if } \ f(t) = 0 \ \text{ and } \ 0 < R_1 < \frac{3S_1}{4} \\ a(t)_1^{iH} - R_1 + \frac{5S_1}{4}, & \text{if } \ f(t) = 0 \ \text{ and } \ \frac{3S_1}{4} \le R_1 < S_1 \\ a(t)_1^{iH} - R_1 - \frac{S_1}{4}, & \text{if } \ f(t) = 1 \ \text{ and } \ 0 < R_1 < \frac{S_1}{4} \\ a(t)_1^{iH} - R_1 + \frac{3S_1}{4}, & \text{if } \ f(t) = 1 \ \text{ and } \ \frac{S_1}{4} \le R_1 < S_1 \end{cases} \tag{10}$$

where $R_1 = \mathrm{mod}(a(t)_1^{iH}, S_1)$, t denotes the sequence of the wavelet coefficients and $a(t)_1^{i'H}$ are samples forming $A_1^{i'H}$ as

$$A_1^{i'H} = \{a(t)_1^{i'H}, 1 \le t \le \frac{L_1}{2^H}\} \tag{11}$$

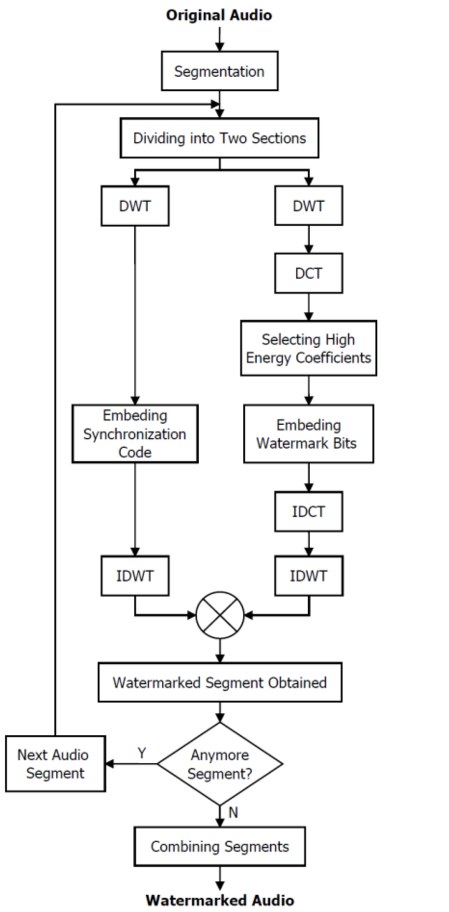


Figure 3: Proposed Audio Watermarking Algorithm

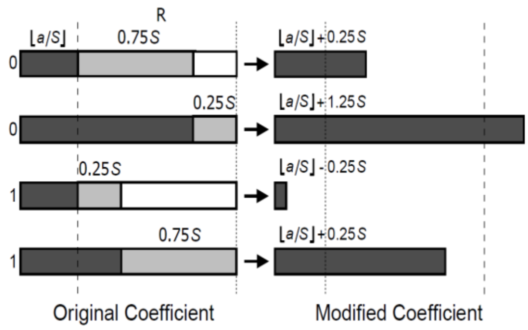


Figure 4: 4-Point Quantization

In (10), the quantization coefficient S_1 is calculated based on the music genre and the required SNR. Then having substituted A_1^{iH} with $A_1^{i'H}$ in (9) $A_1^{i'}$ can be reconstructed using IDWT. This technique improves the stability as well as the transparency of the hidden digital watermark. It should be noted that in similar algorithms using the 2-point quantization [27], the coefficients can be changed up to $\pm 1.5S$, where S represents the quantization coefficients used in (10) and (13). In fact, a coefficient is modified no matter what the remainder of the coefficient divided by S is.

However, in the 4-point quantization method, the modification is based on the remainder of the division in two intervals. As illustrated in Figure 4, this results in a smoother modification and therefore, the resulted watermarked audio shows improved quality and higher SNR.

Watermark Insertion

This section explains how the digital watermark bits are inserted into A_2^i. First, H-level DWT produces coefficients

$$A_2^{iH}, D_2^{iH}, D_2^{i(H-1)}, \ldots, D_2^{i1} \tag{12}$$

from A_2^i. Term A_2^{iH} is an approximation of A_2^i. Then, to make the energy of the audio signal totally concentrated in one section, DCT is applied to A_2^{iH}. In the proposed audio watermarking approach, like the quantization technique described in Section 6.2, a 4-point quantization is used for inserting the digital watermark bits into the high energy DCT coefficients. This is shown as

$$a(t)_2^{i'HC} = \begin{cases} a(t)_2^{iHC} - R_2 + \frac{S_2}{4}, & \text{if} \quad w(t) = 0 \quad \text{and} \quad 0 < R_2 < \frac{3S_2}{4} \\ a(t)_2^{iHC} - R_2 + \frac{5S_2}{4}, & \text{if} \quad w(t) = 0 \quad \text{and} \quad \frac{3S_2}{4} \le R_2 < S_2 \\ a(t)_2^{iHC} - R_2 - \frac{S_2}{4}, & \text{if} \quad w(t) = 1 \quad \text{and} \quad 0 < R_2 < \frac{S_2}{4} \\ a(t)_2^{iHC} - R_2 + \frac{3S_2}{4}, & \text{if} \quad w(t) = 1 \quad \text{and} \quad \frac{S_2}{4} \le R_2 < S_2 \end{cases} \tag{13}$$

where $w(t) = w(i, j)$ with $t = i \times P + j$,

$$A_2^{iHC} = DCT(A_2^{iH}) = \{a(t)_2^{iHC}, 1 \le t \le \frac{L_2}{2^H}\} \tag{14}$$

and $R_2 = \mathrm{mod}(a(t)_2^{iHC}, S_2)$. In (13), S_2 represents the quantization coefficient calculated based on the music genre and the required SNR. Then A_2^{iH} is replaced by $A_2^{i'H}$ in (12) and therefore, $A_2^{i'}$ can be reconstructed using IDWT. In addition, $A_2^{i'H}$ is calculated using IDCT as

$$A_2^{i'H} = IDCT(A_2^{i'HC}) \tag{15}$$

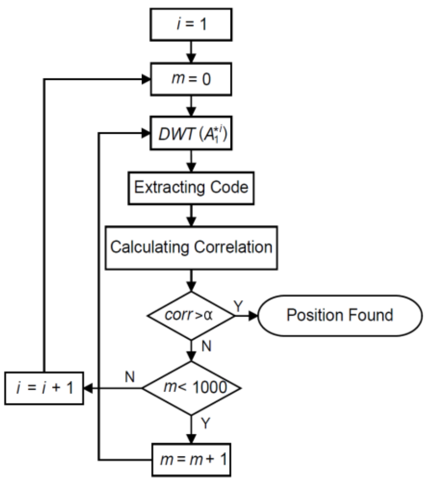


Figure 5: Synchronization Process

In (15), $A_2^{i'HC}$ is defined as

$$A_2^{i'HC} = \begin{cases} a(t)_2^{i'HC}, & \text{if} \quad 1 \le t \le P \times Q \\ a(t)_2^{iHC}, & \text{if} \quad (P \times Q) + 1 \le t \le \frac{L_2}{2^H} \end{cases} \tag{16}$$

Watermark Detection

Using the proposed audio watermarking algorithm, there is no need to the original (non-watermarked) audio when retrieving the watermark bits from the host audio. Instead, the synchronization process needs to be carried out on the host audio signal prior to extracting the watermark. This process consists of finding the exact location of the synchronization code in the watermarked signal followed by extracting the synchronization code bits.

1) Synchronization Process: As shown in Figure 5, the process is carried out on section one of consecutive segment of the host audio, A_1^{*i} with i counting from 1. Symbol "*" reminds that the process is performed on the watermarked audio signal. As indicated in the figure, in every iteration, section A_1^{*i} is selected and is processed from the $(N+m)$-th sample onwards, where m counts from 0 to 1000 and $N=(i-1)\times(L_1+L_2)+1$. The Synchronization process follows as

- Having set $i=1$ and $m=0$, the process starts from the first sample of section A_1^{*i}.
- Like the insertion process described in Section 6.2, A_1^{*i} is sent to a DWT block to calculate the coefficients as $A_1^{*iH}, D_1^{*iH}, D_1^{*i(H-1)}, \ldots, D_1^{*i1}$.
- Using A_1^{*iH} defined as

$$A_1^{*iH} = \{a(t)_1^{*iH}, 1 \le t \le \frac{L_1}{2^H}\} \tag{17}$$

bits of a potential synchronization code are extracted as

$$F^* = f^*(t) = \begin{cases} 0, & \text{if} \quad \mathrm{mod}(a(t)_1^{*H}, S_1) \le \dfrac{S_1}{2} \\ 1, & \text{if} \quad \mathrm{mod}(a(t)_1^{*H}, S_1) > \dfrac{S_1}{2} \end{cases} \tag{18}$$

- For that specific sample, the correlation between the extracted and the original code is calculated.
- The above is repeated for another 1000 samples $(1 \le m \le 1000)$ of section A_1^{*i} until the sample corresponding to the maximum correlation is found as shown in Figure 5.
- If not found, the previous steps are repeated for the next segment by incrementing i by one and resetting m m to 0.
- Once found, sample $(I-1)\times(L_1+L_2)+1+M$ (i.e. $i=I$ and $m=M$) is where the synchronization code can be extracted from. The segment corresponding to the location is called A^{*I} (i.e. $i=I$).

In abstract, section A_1^{*i} is shifted sample by sample and the correlation is calculated. If the correlation is less than a threshold, the process is repeated for section one of the next segment.

2) Watermarking Bit Extraction: Having successfully carried out the synchronization process, the starting point of the watermark bits in the host audio can be determined as follows.

- Section A_2^{*I} with the length of L_2 is selected from the watermarked audio signal.
- DWT is applied to the section and consequently the DWT coefficients $A_2^{*IH}, D_2^{*IH}, D_2^{*I(H-1)}, \ldots, D_2^{*I1}$ are resulted.

- Then, passes A_2^{*IH} through a DCT block and is calculated as

$$A_2^{*IHC} = \begin{cases} a(t)_2^{*IHC}, & \text{if} \quad 1 \le t \le P \times Q \\ a(t)_2^{*IHC}, & \text{if} \quad (P \times Q) + 1 \le t \le \dfrac{L_2}{2^H}. \end{cases} \tag{19}$$

- The watermark bits are extracted from the DCT coefficients as

$$W' = w'(t) = \begin{cases} a(t)_2^{*IHC}, & \text{if} \quad \text{mod}(a(t)_2^{*IH}, S_2) \le \dfrac{S_2}{2} \\ a(t)_2^{*IHC}, & \text{if} \quad \text{mod}(a(t)_2^{*IH}, S_2) \le \dfrac{S_2}{2}. \end{cases} \tag{20}$$

- Arnold transform reconstructs the binary watermark picture from the extracted bits.

PERFORMANCE METRICS AND EVALUATION PREPARATION

As described in Section 1, many different audio watermarking schemes have been proposed in the literature, however, it is very hard to find a common set of criteria developed for evaluating the performance of different audio watermarking approaches. In this manuscript, performance of the audio watermarking algorithm is evaluated with respect to the robustness [32] and the imperceptibility (inaudibility/clearness).

Robustness

In the audio watermarking area, the concept of robustness is not clearly defined and there is a doubt whether a straight forward definition can ever be developed. However, researchers agree that BER (bit error rate) calculated as

$$BER = \frac{E}{P \times Q} 100\% \tag{21}$$

is one the key measures [32]. In (21), E is the number of erroneous bits of the watermark (picture) during the retrieval process. BER expresses the difference between the watermark bits embedded in the host audio signal, and the watermark bits extracted at the receiver side. In other words, BER is error probability of single bit that extracted from watermarked signal. However, in some cases where the watermark represents the signature of the author or the copyright owner [32], NC (normal correlation) calculated as

$$NC = \frac{\sum_{i=1}^{P} \sum_{j=1}^{Q} w(i,j) w'(i,j)}{\sqrt{\sum_{i=1}^{P} \sum_{j=1}^{Q} w^2(i,j)} \sqrt{\sum_{i=1}^{P} \sum_{j=1}^{Q} w'^2(i,j)}} \tag{22}$$

can be a more sensible measure for expressing the robustness of the audio watermarking algorithm. In fact, by setting a threshold value for NC, the receiver can decide whether the extracted watermark correlates (is similar) with the signature.

Imperceptibility

In any audio watermarking method, inserting watermark information introduces a small amount of distortion to the host audio signal. Therefore, the quality (clearness) of the watermarked audio can be used as another criterion for evaluating the performance of audio watermarking algorithms. Having considered the difference between the magnitudes of the original and the watermarked audio signals as noise, SNR calculated as

$$SNR(A,\tilde{A}) = 10\log_{10}\frac{\sum_{n=1}^{N} a^2(n)}{\sum_{n=1}^{N}(a(n)-\tilde{a}(n))^2}\ \text{db} \tag{23}$$

refers to the amount of noise that the algorithm adds to the audio signal. In (23), A represents an N-sample audio signal while $\tilde{A}$ is the watermarked version of that signal.

Simulation Preparation

To find the robustness of the proposed audio watermarking algorithm, it is simulated with 16-bit 44100 Hz audio signals in MATLAB 9.0 environment. In addition, a picture with 64×64 pixel is used in the simulation as the watermark along with a 13-bit Barker code +1+1+1+1+1−1−1+1+1−1+1−1+1 as the synchronization code. Sections sizes L_1 and L_2 are set corresponding to the length of the synchronization code, the picture size and the level of DWT. Therefore, for a 3-level DWT, $L_1 = 13 \times 8 = 104$ and $L_2 = 64 \times 64 \times 8 \times 8 = 2^{18}$.

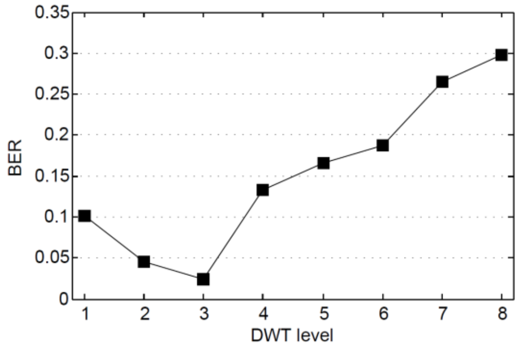


Figure 6: BER versus DWT Level

As a part of the simulation preparation, effects of different levels of DWT on the result of the proposed audio watermarking algorithm are studied. As illustrated in Figure 6, the best tradeoff between robustness and DWT depth is obtained at $H = 3$. Although in the higher levels of DWT the algorithm reaches to higher SNR values, in those levels, the signal's bandwidth that can hold the digital watermark bits becomes narrower and gets closer to lower frequencies of the audio signal. This can lead to instability in the algorithm. Before the simulation starts, the quantization coefficients S_1 and S_2 have to be determined. Increasing these values results in improving the robustness of the audio watermarking algorithm against the attacks, however, this can cause a decline in SNR. Therefore, selecting the quantization coefficients is a tradeoff between robustness and SNR.

As noted before in Sections 6.2 and 6.3, the music genre has a very important role when determining values for the quantization coefficients. This means that for every audio signal, quantization coefficients must be specifically tuned based on the style and the music genre. As shown in Figure 7, for the same BER, a classic music with high energy (Audio1) shows a higher SNR compared to that of a pop music with low energy (Audio2). As another issue, when selecting S_1 and S_2, this fact should be taken into consideration that since most bits embedded in the audio signal correspond to the digital watermark, S_2 has a greater contribution to SNR than that of S_1. In fact, as appeared in Figure 7, the effect of S_1 on SNR becomes noticeable only when $S_2 < 0.1$, however, increasing S_1 to values greater than 1 introduces noise between blocks of the watermarked audio. On the other hand, according to IFPI (the International Federation of the

Phonographic Industry), an audio watermarking algorithm can only degrade the perception of the original audio signal greater than 20 dB (ie SNR > 20 db) [33]. Therefore for the performance evaluation, $S_1 = 0.5$ and $S_2 = 0.35$ are selected that lead to SNR = 28.6 db.

In addition, as appeared in Figure 8, robustness of an audio watermarking algorithm decreases by increasing the value of S_2. This means that when determining the quantization coefficients, the desired robustness must be taken into consideration as well. However, since a specific audio watermarking technique is expected to show different resistances against different types of attacks, the quantization coefficients may have to be altered once circumstances change. This is shown in Figure 8 and is demonstrated in detail in the next sections.

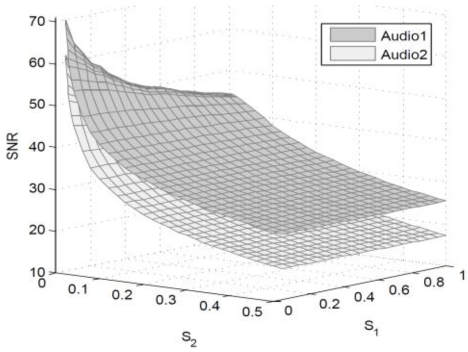


Figure 7: SNR versus S_1 and S_2

EVALUATION RESULTS

In this section, results of a number of simulations carried out on the proposed audio watermarking algorithm are presented. In the presented robustness evaluation procedure, first, different types of attacks are applied to the host audio signal. Then, for every attack, the watermark picture is extracted from the manipulated host audio and finally, the robustness metrics discussed in Sections 8.1 are calculated and listed in Table 1.

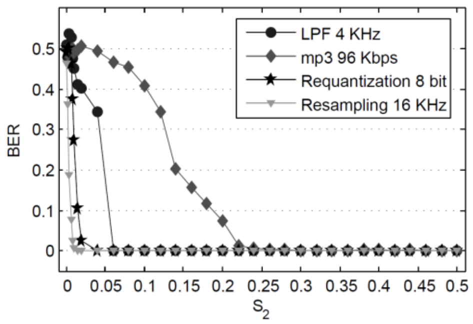


Figure 8: BER versus S_2

In addition, the above procedure is carried out on three major audio watermarking algorithms as

- the method using only DWT for watermark insertion (algorithm A),
- the algorithm using only DCT for watermark insertion (algorithm B), and
- the approach reported in [27] (algorithm C),

and the robustness metrics are listed in Table 1 and are compared with those of the proposed audio watermarking algorithm. The attacks selected for the simulations are resampling, low pass filtering (LPF), MPEG layer 3 (mp3) compression, additive noise, requantization and cropping. To make a fair comparison, the optimum values reported in the literature are used for calculating SNR of algorithms A, B and C that are 25.4 db, 26.9 db and 27.9 db, respectively. These values are all less than that of the proposed audio watermarking scheme however, they are very close to 28.6 db and definitely satisfy the 20 db IFPI requirements.

Resampling

In the resampling attack, the original sampling frequency of the watermarked audio is changed to a lower frequency. This results in a decrease in the bandwidth of the audio signal. Therefore, if the watermark is already embedded in the high frequency components of the audio signal, the watermark can be destroyed. As appeared in rows 2 to 6 of Table 1, except for 32 KHz, which is a very close sampling frequency to that of the original audio signal (44.1 KHz), the proposed algorithm shows a better robustness in terms of BER and NC compared to those of algorithms A and C. The reason is that in the proposed method, the watermark is inserted in the high energy coefficients of DCT that correspond to the low frequency component of the host audio (most of the energy of an audio signal is in the low frequencies). Algorithm B embeds the watermark bits in DCT coefficients and since DCT compresses the energy of the audio signal in the coefficients, the resulting watermarking process affects high energy component of the audio signal and therefore, as indicated in the table, algorithm B looks as robust as the proposed algorithm against the resampling attacks with sampling frequencies of 16 and 8 KHz.

Low Pass Filtering

Low pass filtering, one of the most common attacks performed on watermarked audio files, removes the high frequency component of the audio signal. As shown in Table 1, in every row from 7 to 10, BER of the proposed algorithm is less than those of algorithms A, B and C. The same relation can be seen for NC as well. As discussed in Section 6.3, the proposed algorithm embeds the watermark in the low frequency component of the host audio signal and therefore, low pass filtering has not affect on the inserted picture.

Table 1: Robustness Evaluation Results RS, AN, RQ, SD and CR Denote Resampling, Additive Noise, Requantizing, Standard Deviation and Cropping, Respectively

Attack Type	Proposed Algorithm			DWT Only Algorithm (A)		DCT Only Algorithm (B)		Algorithm in [27] (C)	
	NC	BER	Watermark	NC	BER	NC	BER	NC	BER
No Attack	1.00000	0.00000	LOGO	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
RS-32 KHz	1.00000	0.00000	LOGO	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
RS-16 KHz	1.00000	0.00000	LOGO	0.91431	0.03223	1.00000	0.00000	0.99867	0.00049
RS-8 KHz	1.00000	0.00000	LOGO	0.73613	0.10547	0.99003	0.00366	0.70545	0.12134
RS-4 KHz	1.00000	0.00000	LOGO	Failed	Failed	0.52885	0.20386	0.68190	0.13403
RS-2.5 KHz	1.00000	0.00000	LOGO	Failed	Failed	Failed	Failed	0.51211	0.21973
LPF-16 KHz	1.00000	0.00000	LOGO	0.58379	0.17651	0.75902	0.09668	0.73200	0.10889
LPF-10 KHz	1.00000	0.00000	LOGO	Failed	Failed	0.93393	0.02490	0.95993	0.01489
LPF-8 KHz	1.00000	0.00000	LOGO	Failed	Failed	0.67024	0.13550	0.65398	0.14575
LPF-4 KHz	1.00000	0.00000	LOGO	Failed	Failed	0.60036	0.17090	0.61771	0.16406
mp3- 256 Kbps	1.00000	0.00000	LOGO	0.83471	0.04202	1.00000	0.00000	1.00000	0.00000
mp3-128 Kbps	1.00000	0.00000	LOGO	0.65323	0.15006	0.86028	0.05420	0.64768	0.14746
mp3-96 Kbps	0.93209	0.03101	LOGO	Failed	Failed	0.42674	0.25781	0.22337	0.36816
mp3-64 Kbps	0.49743	0.24072	LOGO	Failed	Failed	Failed	Failed	Failed	Failed
AN-SD=0.005	1.00000	0.00000	LOGO	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
AN-SD=0.01	1.00000	0.00000	LOGO	0.99335	0.00244	0.97316	0.01001	0.93467	0.02441
AN-SD=0.02	1.00000	0.00000	LOGO	0.68971	0.12988	0.52458	0.20923	0.40849	0.26270
RQ-8 bit	1.00000	0.00000	LOGO	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
CR (5×100)	1.00000	0.00000	LOGO	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
CR (10×1000)	1.00000	0.00000	LOGO	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000

MPEG Layer 3 Compression

Resisting compression attacks is a serious challenge for audio watermarking algorithms. Among this group of attacks, mp3 compression is mainly used for illegally distributing copies of music files over the Internet. In the robustness evaluation, mp3 compression rates from 256 Kbps to 64 Kbps are applied to the host audio signal. As indicated in rows 11 to 14 of Table 1, all four algorithms can be considered robust against mp3 256 Kbps attack, however, by increasing the compression rate, the robustness of all algorithms declines. In mp3 compression, positions of the audio samples are relocated and therefore, the synchronization bits embedded in the time domain are destroyed (the watermark position is lost). However, since the proposed algorithm embeds the synchronization bits in the wavelet coefficients, it shows better robustness compared to those of the other three algorithms.

Additive Noise

This type of attack tries to destroy the watermark by adding small amount of low energy noise to the host audio signal. Rows 16 and 17 of Table 1 show that the proposed algorithm has better robustness than those of the other three algorithms since the watermark bits are embedded into high energy component of the audio signal and therefore, the low energy additive noise is not able to corrupt the watermark information.

Requantization

The standard quantization bit length for CD quality of audio (music) files is 16 bit. In requantization attack, the quantization bit length of audio is decreased to values as small as 8 bit. Requantization process decreases the dynamic range of the audio samples without modifying the overall shape and the frequency specification of the audio signal. Therefore all four algorithms look robust against the requantization attack.

Cropping

Cropping attack tries to fail the watermark extraction phase by removing pieces of the host audio signal that contain synchronization data. In this evaluation the watermarked signal is cropped randomly; 100 and 1000 samples from 5 and 10 randomly selected positions, respectively. Therefore, in these examinations, a total of 500 and 10000 samples are removed from the host audio, consecutively. As shown in the last rows of Table 1, all algorithms are robust against cropping attack. In the proposed scheme, the watermark is hidden repeatedly all over the length of the audio signal and also the synchronization bits are embedded in wavelet domain. Therefore, the bits have a chance to be extracted from parts, where are not cropped.

CONCLUSIONS

Distributing copyrighted audio materials over the Internet has become a real problem for the music industry. An effective approach challenging the piracy is audio watermarking. In this manuscript a novel algorithm for embedding and extracting a digital picture as the watermark into a digital audio signal is presented. In the new algorithm, the audio signal is divided into equal length segments while each segment consists of two sections. To embed the watermark bits, the synchronization bits are inserted into the DWT coefficients of the first section and then, DWT followed by DCT are applied to the second section of the host audio segment.

Now the watermark bits can be embedded into the DCT coefficients. The embedding process takes advantage of a 4-point quantization. The performance evaluation results show that the new scheme not only keeps the introduced noise in a very satisfying level (SNR = 28.6 db > 20 db), but also effectively resists the most common attacks designed for destroying the watermark embedded inside the host digital audio.

REFERENCES

1. H. J. Kim, Y. H. Choi, J. W. Seok, and J. W. Hong, “Audio watermarking techniques,” *Intelligent Watermarking Techniques,* pp. 185-218, 2004.
2. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for data hiding,” *IBM Systems Journal,* vol. 35, no. 3/4, pp. 313–336, 1996.
3. K. P. Shine and S. K. Kumar, “Extended bipolar echo kernel for audio watermarking,” in *Proceedings of International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom ’09)*. IEEE Computer Society, 2009, pp. 487–489.
4. D. Kirovski and H. Malvar, “Spread-spectrum watermarking of audio signals,” *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1020–1033, 2003.
5. H. Malik, S. Khokhar, and A. Rashid, “Robust audio watermarking using frequency selective spread spectrum theory,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’04),* vol. 5, May 2004, pp. V385–V388.
6. E. Tavakoli and M. Tabandeh, “A new method for using a psychoacoustic model with patchwork audio watermarking in dft domain,” in *Proceedings of the 34th Annual Conference of IEEE Industrial Electronics (IECON 2008),* Nov. 2008, pp. 1822–1827.
7. N. Kalantari, M. Akhaee, S. Ahadi, and H. Amindavar, “Robust multiplicative patchwork method for audio watermarking,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1133–1141, 2009.
8. S. Esmaili, S. Krishnan, and K. Raahemifar, “Audio watermarking time frequency characteristics,” *Canadian Journal of Electrical and Computer Engineering*, vol. 28, no. 2, pp. 57–61, 2003.
9. M. Pooyan and A. Delforouzi, “Adaptive and robust audio watermarking in wavelet domain,” in *Proceedings of Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2007),* vol. 2, Nov. 2007, pp. 287–290.
10. N. Kalantari, S. Ahadi, and A. Kashi, “A robust audio watermarking scheme using mean quantization in the wavelet transform domain,” in *Proceedings of 2007 IEEE International Symposium on Signal Processing and Information Technology*, 2007, pp. 198–201.
11. Yeo and H. Kim, “Modified patchwork algorithm: A novel audio watermarking scheme,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 4, pp. 381–386, 2003.
12. W. Yongqi and Y. Yang, “A synchronous audio watermarking algorithm based on chaotic encryption in DCT domain,” in *Proceedings of International Symposium on Information Science and Engineering (ISISE ’08)*, vol. 2, Dec. 2008, pp. 371–374.
13. N. Cvejic and T. Seppanen, *Digital audio watermarking techniques and technologies: applications and benchmarks*. Information Science Publishing, 2007.
14. L. Li, J. Hu, and X. Fang, “Spread-spectrum audio watermark robust against pitch-scale modification,” in *Proceedings of 2007 IEEE International Conference on Multimedia and Expo*, 2007, pp. 1770–1773.

15. D. Kirovski and H. Malvar, "Spread-spectrum audio watermarking: requirements, applications, and limitations," in *Proceedings of 2001 IEEE Fourth Workshop on Multimedia Signal Processing*, Oct. 2001, pp. 219–224.

16. H. Malvar and D. Florencio, "Improved spread spectrum: a new modulation technique for robust watermarking," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 898–905, 2003.

17. N. Cvejic and T. Seppanen, "Spread spectrum audio watermarking using frequency hopping and attack characterization," *Signal Processing*, vol. 84, no. 1, pp. 207–213, 2004.

18. S. shiaw Kuo, J. Johnston, W. Turin, and S. Quackenbush, "Covert audio watermarking using perceptually tuned signal independent multiband phase modulation," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, vol. 2, 2002, pp. 1753–1756.

19. C. Xue-Song, Z. Xuan, G. Chang, and Y. Yong-Tian, "Audio watermarking system based on detection theory of chaotic phase transformation," in *Proceedings of Chinese Control and Decision Conference (CCDC '09)*, pp. 4020–4025.

20. K. Imabeppu, D. Hamada, and M. Unoki, "Embedding limitations with audio-watermarking method based on cochlear-delay characteristics," in *Proceedings of Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '09)*, 2009, pp. 82– 85.

21. M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT-domain system for robust image watermarking," *Signal Processing*, vol. 66, no. 3, pp. 357–372, 1998.

22. H. Zhang, L. Cai, X. Meng, X. Xu, X. Yang, X. Shen, and G. Dong, "Image watermarking based on an iterative phase retrieval algorithm and sine-cosine modulation in the discrete-cosine-transform domain," *Optics Communications*, vol. 278, no. 2, pp. 257–263, 2007.

23. V. Aslantas, S. Ozer, and S. Ozturk, "Improving the performance of DCT- based fragile watermarking using intelligent optimization algorithms," *Optics Communications*, vol. 282, no. 14, pp. 2806–2817, 2009.

24. Z. Lin, J. He, X. Tang, and C.-K. Tang, "Fast, automatic and fine-grained tampered jpeg image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009.

25. M. Vetterli and J. Kovacevic, *Wavelets and subband coding*. Prentice Hall, Englewood Cliffs, NJ, 1995.

26. N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

27. X.-Y. Wang and H. Zhao, "A novel synchronization invariant audio watermarking scheme based on DWT and DCT," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4835–4840, Dec. 2006.

28. Al-Haj, L. Bata, and A. Mohammad, "Audio watermarking using wavelets," in *Proceedings of First International Conference on Networked Digital Technologies (NDT '09)*, 2009, pp. 398–403.

29. L. Kun, Z. Ze, and X. Yanhong, "An algorithm of resist cropping robust audio watermark based on wavelet transformation," in *Proceedings of 2nd International Congress on Image and Signal Processing (CISP '09)*, Oct. 2009, pp. 1–5.

30. X.-H. Ma, Z.-J. Liang, and F.-L. Yin, “A digital audio watermarking scheme based on independent component analysis and singular value decomposition,” in *Proceedings of 2006 International Conference on Machine Learning and Cybernetics*, 2006, pp. 2434–2438.

31. Ukil, “Low autocorrelation binary sequences: Number theory-based analysis for minimum energy level, barker codes,” *Digital Signal Processing*, vol. 20, no. 2, pp. 483–495, 2010.

32. J. Gordy and L. Bruton, “Performance evaluation of digital audio watermarking algorithms,” in *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems*, vol. 1, 2000.

33. S. Katzenbeisser and F. A. P. Petitcolas, *Information hiding techniques for steganography and digital watermarking*. Artech House, Boston, 2000.