Advances in Technology Innovation, vol. 5, no. 1, 2020, pp. 18-32

Frame-synchronous Blind Audio Watermarking for Tamper Proofing and Self-Recovery

Hwai-Tsu Hu*, Ying-Hsiang Lu

Department of Electronic Engineering, National I-Lan University, Yilan, Taiwan

Received 25April 2019; received in revised form 28 May 2019; accepted 22 August 2019

DOI: https://doi.org/10.46604/aiti.2020.4138

Abstract

This paper presents a lifting wavelet transform (LWT)-based blind audio watermarking scheme designed for tampering detection and self-recovery. Following 3-level LWT decomposition of a host audio, the coefficients in selected subbands are first partitioned into frames for watermarking. To suit different purposes of the watermarking applications, binary information is packed into two groups: frame-related data are embedded in the approximation subband using rational dither modulation; the source-channel coded bit sequence of the host audio is hidden inside the 2nd and 3rd -detail subbands using 2^N-ary adaptive quantization index modulation. The frame-related data consists of a synchronization code used for frame alignment and a composite message gathered from four adjacent frames for content authentication. To endow the proposed watermarking scheme with a self-recovering capability, we resort to hashing comparison to identify tampered frames and adopt a Reed–Solomon code to correct symbol errors. The experiment results indicate that the proposed watermarking scheme can accurately locate and recover the tampered regions of the audio signal. The incorporation of the frame synchronization mechanism enables the proposed scheme to resist against cropping and replacement attacks, all of which were unsolvable by previous watermarking schemes. Furthermore, as revealed by the perceptual evaluation of audio quality measures, the quality degradation caused by watermark embedding is merely minor. With all the aforementioned merits, the proposed scheme can find various applications for ownership protection and content authentication.

Keywords: Blind audio watermarking, lifting wavelet transform, 2^N-ary adaptive quantization modulation, rational dither modulation, tamper proofing, self-recovery

1. Introduction

In the age of cloud sharing and mobile access, digital resources (such as speech, image, audio and video files) on the Internet keep increasing dramatically in recent years. Ironically, owing to the availability of convenient computer software, tampering multimedia data is also rampant nowadays. Protection against intellectual property infringement thus becomes an important issue. Digital watermarking is considered a promising countermeasure to cope with this issue [1-2].

Digital watermarks can be embedded in noise-tolerant multimedia signals to fulfill the goals of content authentication, copyright protection, covert communication, etc. Based on the information required for extraction, watermarking schemes can be divided into non-blind and blind categories. Non-blind schemes require the original image and/or watermark for extraction, whereas blind schemes require neither. Depending on the application scenario, audio watermarks can also be classified as robust or fragile. Robust watermarking is meant to be resilient to modification attempts, whereas fragile watermarking makes the embedded information sensitive to any modifications [3]. Among the audio watermarking schemes developed for content authentication, the early purpose of the embedded watermark was focused on the detection and localization of tampered area [4-6].

* Corresponding author. E-mail address: hthu@niu.edu.tw
Tel.: +886-3-9317343; Fax: +886-3-9369507

A potential application of fragile watermarking in the field of audio processing is the self-recovery technique, which embeds a watermark into the audio itself to combat the tampering situations. The embedded watermark is often a compressed version of the original content generated via data compression and coding schemes. The amount of the watermark that survives the tampering can help the receiver to not only locate the tampering areas but also to recover the lost content with a certain quality. A few self-recovery schemes for image signals have been proposed so far, such as [7-10]. Speech signal self-recovery was also attempted in [11-13]; nonetheless, studies of the self-recovery schemes for audio signals were relatively limited. The method proposed in [14] divides the audio into 4 segments and embeds the feature parameters of every segment into the less significant bits (LSBs) of another randomly selected segment. For this method, self-recovery is feasible only if the LSBs are completely retrievable. By contrast, the method in [15] embeds the control bits for self-recovery in the integer Discrete Cosine Transform (intDCT) domain and then employs a compressive sensing technique to retrieve the tampered intDCT coefficients. Although this method is capable of recovering the audio signal tampered by content replacement attacks, it can only restore the attacked signals up to 0.6 % with acceptable quality. Furthermore, the size of the replaced segment must remain identical to enable tampering detection and signal recovery. In fact, it is more common to encounter a situation where the replaced segment holds a different size. The method in [16] attempts to solve such a size discrepancy problem using a synchronization strategy. However, the adopted synchronization strategy appears oversimplified. When the length of the received audio signal is shorter than that of the original, it simply adds a set of zeros at the end of the audio instead of aligning the signal back to the correct position.

One common drawback of the foregoing audio watermarking schemes for self-recovery is that they all lack the countermeasures to cope with cropping and/or time-shifting attacks. A minor time mismatch can disrupt the watermark extraction for subsequent self-recovery. Motivated by the work done in [14-16], we propose an efficient blind audio watermarking scheme that is capable of achieving tamper proofing and self-recovery in the presence of arbitrary content replacement attacks. The remainder of this paper is organized as follows. Section 2 presents two watermarking schemes designed for attaining frame-synchronous blind audio watermarking in the lifting wavelet domain. Section 3 outlines the procedures used in watermark embedding and extraction. The framework for self-recovery is discussed in Section 4. Section 5 evaluates the proposed scheme in terms of imperceptibility, temper proofing, self-recovery, and processing time. In order to illustrate the advantages of the proposed scheme more clearly, Section 5 also provides a comparative evaluation between the proposed self-recovery scheme and the one in [16]. Finally, conclusions are given in Section 6.

2. LWT-based Watermarking Schemes

Among the transforms used to perform audio watermarking, DWT appears to be the most popular due to its perfect reconstruction and good multi-resolution characteristics. In particular, many DWT-based schemes take advantage of quantization index modulation (QIM) [17] to achieve effective watermark embedding. To reduce computational and memory overhead, we adopted a lifting scheme to implement the DWT in this study. A lifting wavelet transform (LWT) comprises three steps: split, prediction, and update for signal decomposition, and another three steps: update, prediction, and merge are needed for signal reconstruction. The LWT saves computational time and enables frequency localization to overcome the weakness of the traditional wavelet. It is regarded as the second-generation wavelet transform [18].

Fig. 1 presents the procedural flow for watermark generation and embedding. As illustrated in the right branch of Fig. 1, we first apply a 3-level LWT to decompose a host audio signal into one approximation subband and three detail subbands, each corresponding to a specific frequency range. In particular, the Daubechies-8 basis [19] is used as a wavelet function in the process of LWT. Note that audio watermarking is preferably implemented in low-frequency subbands with relatively high intensity, as these subbands are more tolerable to signal alteration with less impairment in perceptual quality. Theoretically, for audio sampled at 44.1 kHz, the 3^{rd} -level approximation subband spans a frequency range from 0 to 2756 ($= 22050 / 2^3$) Hz,

which is suitable for robust watermarking. Hence, in our design the approximation subband is reserved for embedding the crucial information including synchronization code, frame index, and hash data derived from the channel-coded bit stream. The 2^{nd} and 3^{rd} -level detail subbands are used to hide fragile watermarks that is responsible for data authentication and signal recovery.

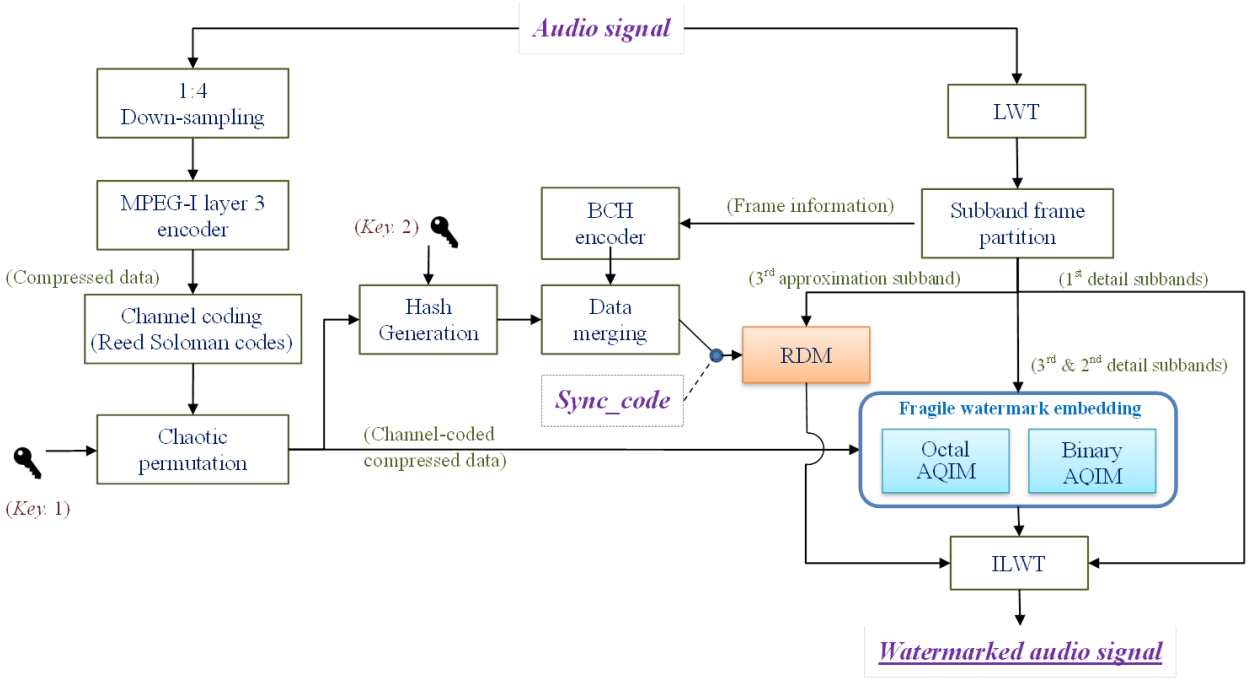


Fig. 1 Watermark generation and embedding

2.1. *Rational dither modulation*

Following the application of LWT to the audio signal, we employ the rational dither modulation [20, 21] to carry out binary embedding in the approximation subband. Let $c_a^{(3)}(n)$ denote the n^{th} coefficient in the 3^{rd} -level approximation subband. By referring to the QIM [17], the embedding of a binary bit $w_a(n) \in \{0,1\}$ into $c_a^{(3)}(n)$ can be formulated as

$$\hat{c}_a^{(3)}(n) = \mathrm{sgn}\left(c_a^{(3)}(n)\right) \times \left(\left\lfloor \frac{\left|c_a^{(3)}(n)\right|}{\Delta_k} - \frac{w_a(n)}{2} + \frac{1}{2} \right\rfloor \Delta_{(n)} + w_a(n)\frac{\Delta_{(n)}}{2} \right) \tag{1}$$

where $\mathrm{sgn}(\cdot)$, $|\cdot|$, $\lfloor \cdot \rfloor$ represent the sign, absolute, and floor functions, respectively. $\Delta_{(n)}$ stands for the step size for quantizing coefficients. The subscript 'a' in the symbol $w_a(n)$ implies the targeted approximation subband wherein the watermark bit is embedded and extracted. The use of the magnitude rather than amplitude in Eq. (1) aims at excluding the trouble of sign flipping. In accordance with the formulation in Eq. (1), the watermarking error, which is defined as the difference between $\hat{c}_a^{(3)}(n)$ and $c_a^{(3)}(n)$, can be assumed to have a uniform distribution over $\left[-\Delta_{(n)}/2, \Delta_{(n)}/2\right]$ with a variance of $\Delta_{(n)}^2/12$. One of the key features of the RDM lies in the acquisition of $\Delta_{(n)}$, which is recursively derivable from previous coefficients through

$$\Delta_{(n)} = \left(\frac{1}{L} \sum_{i=1}^{L} \left(\hat{c}_a^{(3)}(n-i) \right)^2 \right)^{1/2} \times 10^{\frac{F(B_{rep}) - 10\log_{10}(1/12)}{20}} \tag{2}$$

where L stands for the length of the involved coefficients. Similar to the manner in [20, 21], the embedding strength is adaptively controlled at the maximum tolerable level of the human auditory system [22, 23]. The term $10^{\frac{F(B_{rep})}{20}}$ signifies a multiplicative factor for adjusting the embedding strength. $F(B_{rep})$ is the auditory masking threshold in unit of decibel for the Bark scale B_{rep}.

$$F(B_{rep}) = -0.275 \times B_{rep} - 15.025 - \eta \tag{3}$$

with η representing a clearance gap for imperceptibility. B_{rep} can be obtained from a representative frequency f_{rep} using the following empirical formula [24]:

$$B_{rep} = 13\tan^{-1}\left(0.00076 f_{rep}\right) + 3.5\tan^{-1}\left(\left(f_{rep}/7500\right)^2\right) \tag{4}$$

Here we chose the center of the 3^{rd}-level approximation subband as the representative frequency, i.e.

$$f_{rep} = 0.5 \times \frac{1}{2^3} \times \frac{f_s}{2} \tag{5}$$

where f_s is the sampling frequency. Overall, Eq. (2) jointly takes into account of the psychoacoustic features (i.e., $F(B_{rep})$), the quantization error distribution (i.e., $10\log_{10}(1/12)$), and the root-mean-square of previously processed coefficients (i.e., $\{\hat{c}_a^{(3)}(n-i) | i = 1, \cdots, L\}$).

The watermark extraction in RDM requires the derivation of quantization step $\tilde{\Delta}_{(n)}$ based on the same formula presented in Eq. (2). Subsequent to the acquisition of the 3rd-level approximation coefficient $\tilde{c}_a^{(3)}(n)$, the bit $\tilde{w}_a(n)$ residing in $\tilde{c}_a^{(3)}(n)$ can be determined by

$$\tilde{w}_a(n) = \mathrm{mod}\left(\left\lfloor \frac{\left|\tilde{c}_a^{(3)}(n)\right|}{\tilde{\Delta}_{(n)}} 2 + 0.5 \right\rfloor, 2\right) \tag{6}$$

where $\mathrm{mod}(x, y)$ denotes the modulo operation, which returns the remainder after the division of x by y. The tilde symbol atop a participating variable implicates the effect due to possible attacks.

2.2 2^N-ary adaptive quantization index modulation

Analogous to RDM, the 2^N-ary adaptive quantization index modulation (AQIM) modifies the coefficient magnitude according to a 2^N-ary number $w_d(n) \in \{0, 1, \cdots, 2^N - 1\}$.

$$\hat{c}_d^{(j)}(n) = \mathrm{sgn}\left(c_d^{(j)}(n)\right) \times \left(\max\left\{0, \left\lfloor \frac{\left|c_d^{(j)}(n)\right|}{\Delta_k} - \frac{w_d(n)}{2^N} + \frac{1}{2} \right\rfloor \Delta_k \right\} + w_d(n)\frac{\Delta_k}{2^N} \right) \tag{7}$$

where $\max\{\cdot\}$ denotes the maximum value drawn from a set of data. $c_d^{(j)}(n)$ is the n^{th} coefficient in the j^{th}-level detail subband. The floor function within the above equation may, however, render a negative value that is illegitimate to the definition of a magnitude. In case a negative outcome occurs, we simply replace the negative value with zero.

In contrast to the case in RDM, the quantization step Δ_k is computed from the energy of all the coefficients in a frame indexed by an integer k. Given that the watermarking errors maintain a power ratio Γ in decibels, the relationship betwee Δ_k and Γ can be mathematically expressed as follows:

$$10^{\frac{\Gamma}{10}} = \frac{\frac{1}{L_f}\sum_{i=0}^{L_f-1}\left(c_d^{(j)}(i)\right)^2}{\mathrm{E}\left[\frac{1}{L_c}\sum_{i=0}^{L_c-1}\left(\hat{c}_d^{(j)}(i)-c_d^{(j)}(i)\right)^2\right]} = \frac{\frac{1}{L_f}\sum_{i=0}^{L_f-1}\left(c_d^{(j)}(i)\right)^2}{\Delta_k^2/12} \tag{8}$$

where L_f is the frame length. L_c denotes the number of the coefficients involved in the quantization. Basically, $L_c < L_f$. By referring to Eqs. (3) and (4), the value of Γ can also be estimated as

$$\Gamma = -F(B_{rep}) + 10\log_{10}(1/12) \tag{9}$$

In [20, 25-27], it was demonstrated that the quantization step size can be adaptively retrieved from a watermarked audio as long as the energy level remains unchanged throughout the watermarking process. The modification on $c_d^{(j)}(n)$ in Eq. (7) inevitably cause variations in energy, which makes the retrieved Δ_k different from the one used in watermark embedding. Thus, the recovered watermark bits may become inaccurate. This conflict can be settled by first minimizing the overall energy variation of the first L_c coefficients and then tuning the other coefficients in the range between L_c+1 and L_f. More specifically, we first sort the coefficient magnitudes, termed $\rho(l_i) = \left|c_d^{(j)}(l_i)\right|$, in descending order:

$$\hat{\rho}(l_0) \geq \hat{\rho}(l_1) \geq \cdots \geq \hat{\rho}(l_{i-1}) \geq \hat{\rho}(l_i) \geq \cdots \geq \hat{\rho}(l_{L_c-1}) \tag{10}$$

where l_i, which is drawn from $\{0,1,\cdots,L_c-1\}$, signifies the index associated the i^{th} largest magnitude. When applying Eq. (7) to the l_i^{th} coefficient, the optimal solution $\eta_1(l_i)$ is

$$\eta_1(l_i) = \hat{\rho}(l_i) = \left|\hat{c}_d^{(j)}(l_i)\right| \tag{11}$$

and the suboptimal $\eta_2(l_i)$ becomes

$$\eta_2(l_i) = \begin{cases} \hat{\rho}(l_i) - \Delta_k, & \text{if} \quad \left(\hat{\rho}(l_i) > \left|\hat{c}_d^{(j)}(l_i)\right|\right) \,\&\, \left(\hat{\rho}(l_i) > \Delta_k\right) \\ \hat{\rho}(l_i) + \Delta_k, & \text{otherwise.} \end{cases} \tag{12}$$

In general, coefficients with large magnitudes contribute more variations in energy. To minimize the overall energy variation in a frame, we select between $\eta_1(l_i)'s$ and $\eta_2(l_i)'s$ for the coefficient magnitudes in the top L_o ranks.

$$\{\hat{n}_i\} = \arg\min_{\substack{\{n_i \mid i=0,\cdots,L_o-1\} \\ n_i \in \{1,2\}}} \left| \sum_{i=0}^{L_o-1}\left(\eta_{n_i}^2(l_i) - \rho^2(l_i)\right) + \sum_{i=L_o}^{L_\rho-1}\left(\hat{\rho}^2(l_i) - \rho^2(l_i)\right) \right| \tag{13}$$

subject to the constraint that the accumulated energy must be less than the overall energy, i.e.

$$\sum_{i=0}^{L_o-1}\eta_{n_i}^2(l_i) + \sum_{i=L_o}^{L_o-1}\hat{\rho}^2(l_i) \leq \sum_{i=0}^{L_o-1}\left(c_d^{(j)}(l_i)\right)^2 + \sum_{i=L_c}^{L_f-1}\left(c_d^{(j)}(i)\right)^2 = \sum_{i=0}^{L_f-1}\left(c_d^{(j)}(i)\right)^2 \tag{14}$$

The search for $\{\hat{n}_i\}$ in this study is done by a brutal force approach. Thus, the required computation is exponentially proportional to L_o. This study chooses L_o as 8. Substituting $\eta_{\hat{n}_i}(i)$ for the magnitudes in $\left\{\hat{\rho}(l_i) \middle| 0 \leq i \leq L_o-1\right\}$, i.e.

$\hat{\rho}(l_i) \leftarrow \hat{\rho}_\eta(l_i) = \eta_{\hat{n}_i}(l_i)$, yields the least energy variation achievable by the L_o coefficients. Once the magnitude for the l_i^{th} coefficient is determined, the corresponding detail coefficients can be modified as follows:

$$\hat{c}_d^{(j)}(l_i) = \mathrm{sgn}\left(c_d^{(j)}(l_i)\right)\frac{\hat{\rho}_\eta(l_i)}{\rho(l_i)+\varepsilon}; \quad i = 0,1,\cdots,L_c-1; \quad l_i \in \{0,1,\cdots,L_c-1\} \tag{15}$$

where ε represents an infinitesimal number added to the denominator to avoid dividing by zero.

The violation of the constraint (14) implies that the energy collected from the first L_c coefficients exceeds the total amount. It will be impossible to compensate for the excessive portion by regulating the energy over the remaining coefficients, i.e., $\left\{c_d^{(j)}(i) \middle| i = L_c, L_{c+1}, \cdots, L_f - 1\right\}$. In case the inequality (14) cannot maintain after adjusting the first L_o coefficients, we then proceed with the next L_o coefficients in the top ranks, i.e., $\left\{\hat{c}_d^{(j)}(l_i) \middle| i = L_o, \cdots, 2L_o - 1\right\}$, and rerun the adjustment process. Previously altered detail coefficients in the sorted sequence shall remain intact. The adjustment process continues until the constraint shown in Eq. (14) is satisfied. Finally, to ensure a perfect match with the original energy level, we use the remaining $(L_f - L_c)$ coefficients to absorb the energy discrepancy

$$\hat{c}_d^{(j)}(k) = c_d^{(j)}(k)\left(\frac{\sum_{i=0}^{L_f-1}\left(c_d^{(j)}(i)\right)^2 - \sum_{i=0}^{L_c-1}\left(\hat{c}_d^{(j)}(i)\right)^2}{\sum_{i=L_c}^{L_f-1}\left(c_d^{(j)}(i)\right)^2}\right)^{1/2}; \qquad k = L_c, \cdots, L_f - 1. \tag{16}$$

After completing the watermark embedding, we reconstruct the audio by taking the inverse LWT with respect to all subband coefficients. To retrieve the embedded watermark bits from the watermarked audio, we follow the same steps used in the embedding process. The quantization step size $\tilde{\Delta}_k$ can be obtained using Eqs. (8). The i^{th} 2^N-ary number, termed $\tilde{w}_d(i)$, is determined based on the QIM rule:

$$\tilde{w}_d(i) = \mathrm{mod}\left(\left\lfloor \frac{\left|\tilde{c}_d^{(j)}(i)\right|}{\tilde{\Delta}_k} 2^N + 0.5 \right\rfloor, 2^N\right) \tag{17}$$

3. Self-recovery Framework

One of the main features of the proposed watermarking scheme is the self-recovery capability. In order to achieve tamper proofing and self-recovery concurrently, we incorporate the source-channel coding and hashing techniques into the proposed watermarking scheme. The basic idea is to use the frame-partitioned source-channel coded data as the watermark in the embedding phase, and examine the watermark for tamper detection and self-recovery in the extraction phase. In this study, we adopt a MPEG-1 audio layer III codec (termed MP3 for short) to perform a lossy data-compression of the host audio. In consideration of the limited watermarking capacity, the audio signal is encoded at a very low bitrate of 16 kilobits per second (kbps). This is actually achieved by down-sampling the audio by a factor of 4 and then applying the MP3 codec to convert the audio to a bit stream of 64 kbps. The bit stream is further divided into frames of size 2448 (=2×153×8), which can be regarded as 2 message words, each containing 153 bytes.

In this study, Reed-Solomon (RS) codes on the Galois fields $GF(2^8)$ [28] are employed to recover the information destroyed by tampering attempts. For each message word, we use a $(255, 153)$ RS code to form an augmented word of length 255. This arrangement enables the RS code to correct 51 $(=(255-153)/2)$ errors in a row of 255 symbols. In other words, the tolerable tampering rate of the applied RS code is 20% (i.e., 51/255). As a result, the number of bits that were supposed to

embed in each frame is expanded from 2448 to 4080 (=2×255×8). Given that the sampling rate of the audio is 44100 Hz, this amount of binary bits shall be embedded into a frame with its length no less than 44100×2448/16000. Hence we choose the frame length as 6656 samples and embed 4080 bits into the 3rd and 2nd-level detail subbands after performing 3-level LWT decomposition. As shown in Fig. 2, the 3rd and 2nd-level detail subbands respectively consist of 832 and 1664 coefficients in each frame. We tactically embedded 816 octal numbers (3 bits per coefficient) into the 3rd-level detail subband and 1632 binary bits (1 bit per coefficient) into 2nd-level detail subband using the 2^N-ary AQIM discussed in Section 2, thus rendering a total of 4160 bits to accommodate the need of channel-coded data for audio recovery. The extra 80 bits ($=4160-4080$) are reserved for the need of file headers. Also note that we have applied a distinct 2^N-ary AQIM to each detail subband. The use of 8-ary AQIM in the 3rd-level detail subband stems from the consideration that this subband usually contains relatively higher intensity than that found in the 2nd-level detail subband. According to the formula for 2^N-ary AQIM given in Eq. (8), a high energy level also leads to a large quantization step that is supposedly more capable of resisting against malicious attacks.

4. Procedures for Watermark Embedding and Extraction

4.1. Watermark embedding

The procedure for watermarks embedding is detailed in the following.

Step. 1: Apply the 64 kbps MP3 codec to a down-sampled audio signal and convert the output file into a bit stream. Compose the bit sequence as an array of message words with a size of 153 bytes (or equivalently counted as 153 8-bit symbols).

Step. 2: Append the parity symbols to each message word after applying a $(255, 153)$ RS encoder.

Step. 3: The symbols are scrambled among words via the use of $key.1$. This operation allows the RS code to detect and correct symbol errors in the corrupted word.

Step. 4: Divide the message array into groups, each holding 4 consecutive words. For each group,

- Record the frame index as a 16-bit integer and encode this integer using a $(31,16)$ BCH encoder [29].
- Record the total number of frames as a 16-bit integer and encode this integer using a $(31,16)$ BCH encoder.
- Use $key.2$ to randomly permute the symbol sequence in each message word.
- Apply the MD5 hash algorithm [30] to the first 153 symbols in each word and draw 16 hash bits from each hashed output to form a composite hash representation of 64 bits.
- Pack the frame-related information as a bit sequence of length 128, as shown in Fig. 2.

Step. 5: Perform 3-level LWT on the host audio

Step. 6: Partition the coefficients in each subband into frames. For a frame length of 6656 audio samples, there are 832 and 1664 coefficients contained in the 3rd and 2nd-level subbands, respectively.

Step. 7: Distribute the channel-coded symbols obtained in Steps. 3 and 4 to the audio frames. For each frame,

- Embed the synchronization code and frame-related information alternately into the 3rd-level approximation subband.
- Embed 2448 bits into the 3rd-level detail subband using 8-ary AQIM.
- Embed 1632 bits into the 2nd-level detail subband using binary AQIM.

Step. 8: Take inverse LWT to obtain the watermarked audio.

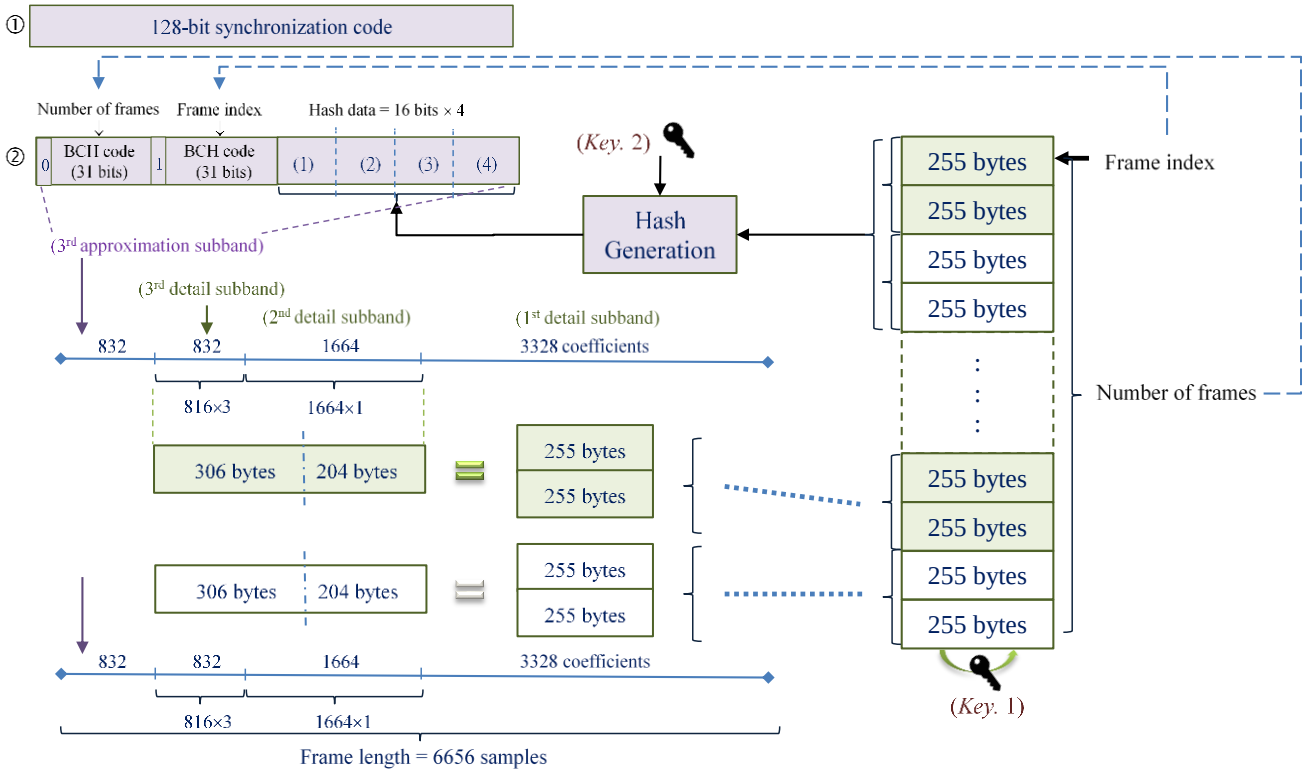


Fig. 2 Arrangement of watermark bits

4.2. Watermark extraction

Fig. 3 outlines the procedure for watermark extraction, tampering detection, and self-recovery. The required steps are outlined as follows:

Step. 1: Conduct 3-level LWT.

Step. 2: Extract the embedded bits from the approximation subband using RDM discussed in Section 2.

Step. 3: Apply a matched filter to the extracted bit sequence. The synchronization code in reverse order serves as the filter coefficients. Given that $\phi(n) \in \{0,1\}$ denotes the synchronization code of length l_{sync}, feeding the extracted $\tilde{w}_a(n)$ into the matched filter results in,

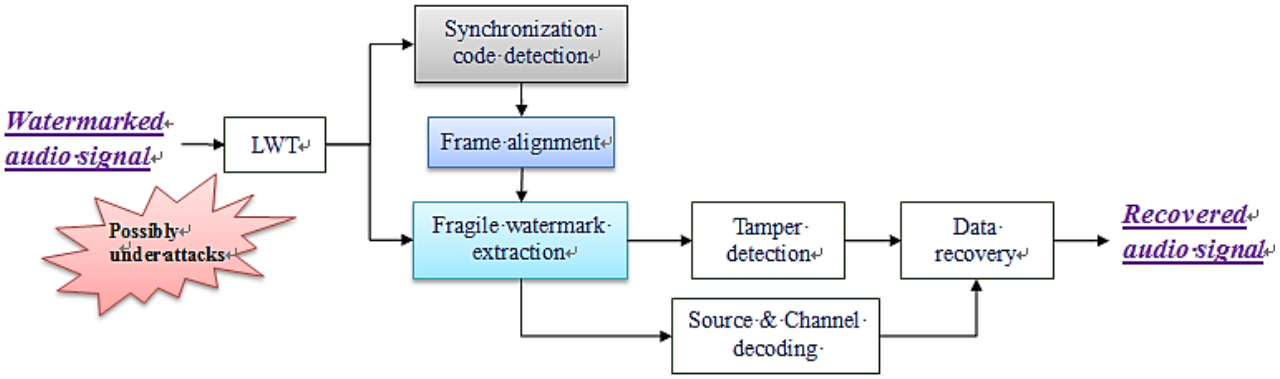


Fig. 3 Block diagram of watermark extraction for tampering detection and audio recovery

$$M(n) = \sum_{i=0}^{l_{sync}-1} \left(2\phi(l_{sync}-1-i)-1\right)\left(2w_a(n-i)-1\right) \tag{18}$$

Ideally, a salient peak of value l_{sync} occurs whenever $\tilde{w}_a(n)$ perfectly matches with the synchronous code.

Step. 4: For each frame,

- Retrieve the frame-related data and the hash bits from two consecutive frames; acquire the number of total frames and frame index using a $(31,16)$ BCH decoder.
- Extract 2448 bits from the 3rd-level detail subband using 8-ary AQIM.
- Extract 1632 bits from the 2nd-level detail subband using binary AQIM.
- Rearrange these (2448+1632) bits as two words, each comprising 255 8-bit symbol (1 byte per symbol).
- Place these two words in the corresponding index entry.

Step. 5: Use $key.2$ to restore the symbol sequence in each message word.

Step. 6: Use $key.1$ to restore the original permutation of the symbol array.

Step. 7: Pass the message word to the RS decoder to obtain the source-coded audio symbols.

Step. 8: Generate the hash bits from each word using the MD5 hash algorithm and compare these bits with those recorded in the approximation subband. If the hash bits are identical, the symbol sequence is assigned to the location indicated by the frame index. Otherwise, the frame is labeled as tampered at the receiver.

Step. 9: Use a MP3 decoder to decompress the audio signal from the extracted watermark bits and up-sample the output by a factor of 4.

Step. 10: If the audio frame has been tampered, then we substitute the up-sampled audio signal for the tampered audio content.

Since the RS decoding process is capable of removing 51 ($=(255-153)/2$) errors in a row of 255 symbols, tampering is recoverable as long as the tampering rate is below 0.2 (=51/255); otherwise, the RS decoder and recovery process fail.

5. Performance Evaluation

The test materials in the following experiments comprised twenty-four 30-second music clips collected from a variety of compact discs, including vocal arrangements and ensembles of musical instruments. The music clips can be classified into four categories: classical (3), pop (7), rock (7), soundtracks (7). All audio signals were sampled at 44.1 kHz with 16-bit resolution. The parameters used in the proposed watermarking scheme were set as follows: $\eta = 2$, $L_0 = 8$, $l_{sync} = 128$; $f_{rep} = 1378.1$, 4134.4, and 8268.8 Hz for the 3rd-level approximation subband, 3rd-level detail subband, and 2nd-level detail subband, respectively; The back-tracing length L used in the RDM was set to 416. $L_c = 816$ and $L_f = 832$ were chosen for the 8-ary AQIM in the 3rd-level detail subband, while $L_c = 1632$ and $L_f = 1664$ were for the binary AQIM in the 2nd-level detail subband.

5.1 Imperceptibility test

The quality of the watermarked audio signal was evaluated using the SNR defined in Eq. (19) along with the perceptual evaluation of audio quality (PEAQ) metric [31].

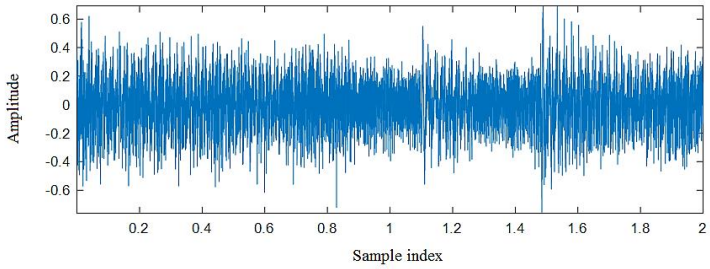
$$SNR = 10\log_{10}\left(\frac{\sum_n s^2(n)}{\sum_n \left(\hat{s}(n)-s(n)\right)^2}\right) \tag{19}$$

where $s(n)$ and $\hat{s}(n)$ denote the original and watermarked audio signals, respectively. The PEAQ simulates the subjective evaluation of human subjects. It renders an objective difference grade (ODG) between -4 and 0, signifying a perceptual impression from "very annoying" to "imperceptible". In this study, the PEAQ metric for the imperceptibility test was an implementation released by the TSP Lab at McGill University [32].

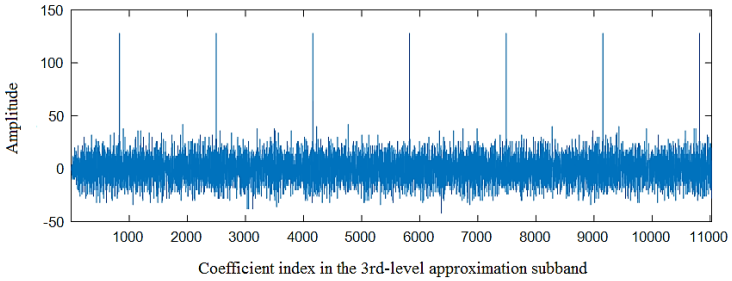
Table 1 summarizes the experiment results with respect to the test materials. For each audio signal of 30 seconds long, there are over 198 frames of size 6656 can be embedded and at least 99 of them contain the synchronization code. Embedding the synchronization codes and frame-related data into the 3rd-level approximation subband rendered an average SNR of 28.34 dB, which led to an average ODG score around -0.30. The subsequent embedding of the channel coded symbols into the 3rd and 2nd -level detail subbands brought the SNR to 27.36 dB and caused the ODG to slightly drop to -0.44. Such a result suggests that the proposed watermarking schemes merely have minor influence on perceptual quality. Moreover, for all audio files in the test, the embedded synchronization codes were perfectly detected using the matched filter. Fig. 4 shows one such example, wherein the peaks with a height of 128 repeatedly appear for every 1664 approximation coefficients.

Table 1 Quality measures of the watermarked audio after applying RDM and $\boldsymbol{2^N}$-ary AQIM to the audio signals

Quality measure		RDM (in 3rd-level app. Subband)	$\boldsymbol{2^N}$ -ary AQIM (in 3rd & 2nd-level detail subbands)	RDM+ $\boldsymbol{2^N}$ -ary AQIM
SNR [dB]	Mean	28.34	35.59	27.36
	Standard deviation	0.25	3.92	0.46
ODG	Mean	-0.30	-0.22	-0.44
	Standard deviation	0.38	0.26	0.43



(a) Audio siganl

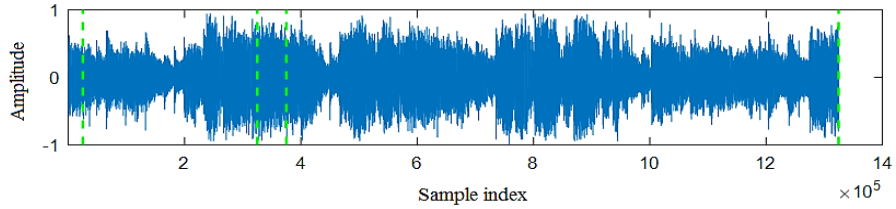


(b) Output of the matched fliter

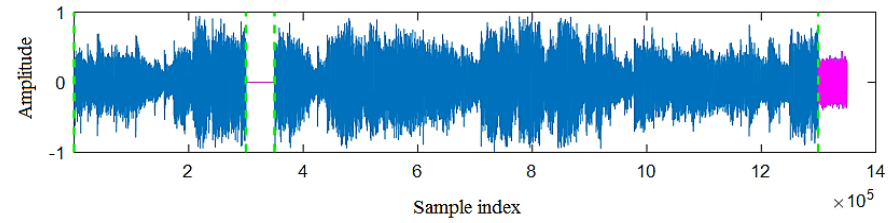
Fig. 4 Matched filtering with respect to the watermark bits obtained by the RDM

5.2 Tamper detection and recovery

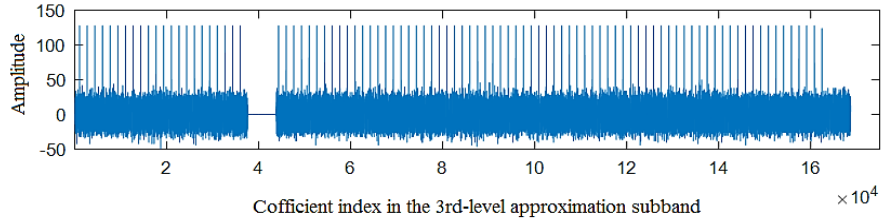
A representative audio signal was employed to demonstrate the competence of the proposed scheme for tampering detection and localization. We conducted three types of attacks (namely, deletion, substitution, and insertion) on the audio signal with the self-recovering watermark embedded. The deletion attack cropped the leading 25000 samples of the watermarked audio signal. The substitution attack replaced the watermarked audio signal with zero over the range between 325001 and 375000. For the insertion attack, we appended 50000 samples of random noise at the end of the watermarked audio signal. Both the substitution and insertion attacks represent possible attempts on counterfeiting the audio signal.



(a) Audio with fragile watermarks embeded

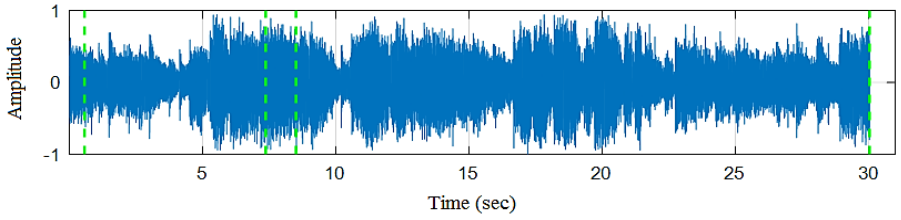


(b) Tampered audio

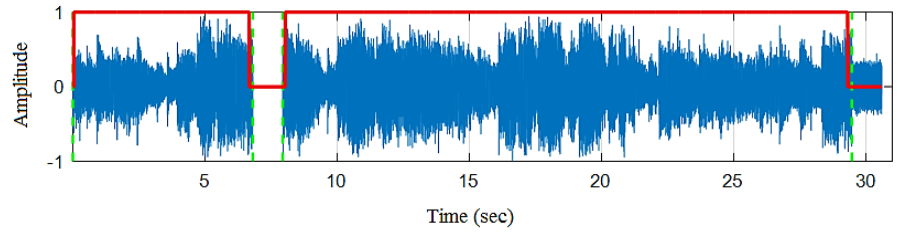


(c) Output of the matched fliter

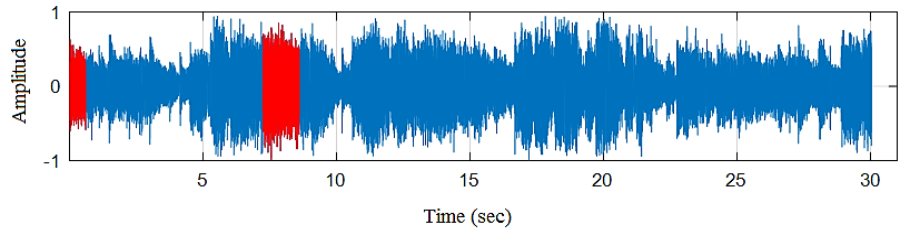
Fig. 5 Illustration of three types of tampering attack



(a) Audio with fragile watermarks embeded



(b) Tampered audio



(c) Recoverd audio

Fig. 6 Illustration of tampering detection and audio signal recovery

Fig.5 (a) and (b) respectively present the original and tampered watermarked audio signals. The watermark bits hidden in the 3rd-level approximation subband are extracted, bipolarized (i.e., $\{0,1\} \rightarrow \{-1,1\}$), and finally fed into a matched filter. Fig.

5 (c) depicts the output of the matched filter. A sharp peak with its magnitude greater than a predefined threshold (e.g., 80) can serve as an indicator to demarcate the frame boundary. The tampered signal is then processed using the watermark extraction and self-recovery procedures discussed in Sections 2-4. More specifically, subsequent to frame synchronization, the hash bits are used to verify the veracity of individual frame content. As shown in Fig. 6(b), a nonzero level (delineated as a bold solid red line) signifies intact frames and a zero level specifies the occurrence of tampering. All the data extracted from the 2nd and 3rd-level detail subbands are then employed to reconstruct an MP3 decompressed version of the audio signal. Eventually, the lost contents in the tampered frames are replaced by the reconstructed ones, which are drawn in red in Fig. 6(c). This typical example demonstrates that our scheme not only accurately locates the tampered audio frames but also possesses a self-recovering capability.

5.3 Processing Time

The proposed self-recovery scheme comprises five basic modules to carry out watermark embedding. The first module involves source-channel encoding and hashing technique jointly used to constitute the bit sequence for audio recovery. As the watermarking is accomplished in three low-to-middle frequency subbands, we need a 3-level LWT and another inverse LWT to decompose and recompose the audio signal. These two transformations are extra computational burdens for the watermarking performed in the LWT domain. The operations situated in between the LWT and ILWT contain two sorts of watermark embedding, namely, the synchronization code sequence in the 3^{rd} level approximation subband and the channel-coded bit stream in both the 2^{nd} and 3^{rd} detail subbands. As for the process of watermark extraction, we only need a 3-level LWT to decompose the watermarked audio. The detection of the synchronization code enables the alignment of frame boundary, which facilitates the watermark retrieval in the 2^{nd} and 3^{rd} detail subbands. Possible errors in the watermark bits shall be amended with the assistance of the RS channel coder. Eventually, the veracity and integrity of the received audio can be authenticated using hashing comparison.

Table 2 Processing time required for each program module in watermark embedding and extraction processes

Program Modules in Watermark Embedding	**Processing Time [sec]**	
	Mean	**Standard deviation**
Conduct (1) Data encoding & hashing (2) Bit arrangement	6.377	0.193
Perform LWT	0.947	0.009
Embed *sync_code* using RDM	0.780	0.019
Embed channel-coded data using AQIM	2.852	0.075
Perform ILWT	0.954	0.008
Overall	**11.910**	**0.258**
Program Modules in Watermark Extraction	**Processing Time [sec]**	
	Mean	**Standard deviation**
Perform LWT	0.957	0.013
Align frames via the detection of sync_code	0.027	0.007
Extract watermark (coded data)	0.029	0.004
Perform channel-decoding and hashing comparison	1.267	0.037
Restore the tampered signal if necessary	-	-
Overall	**2.281**$^{+}$	**0.048**$^{+}$

We implemented the proposed watermarking algorithm in a Matlab environment operating with a 4 GHz Intel(R) Core(TM) i7-4790K CPU and 32 GB RAM. Table 2 lists the average computation time for the twenty-four 30-second audio signals in the test set. In general, it takes 11.91 seconds to complete the watermark embedding for an audio file of 30 second long. Among the five modules in the whole process, the data encoding and hashing consume about 53.54% of the computational time. The actual embedding in LWT subbands requires 5.533 seconds in total. Compared to the lengthy computation required in the embedding process, the time spent on watermark extraction is greatly reduced while extracting watermark bits from the 2^{nd} and 3^{rd} detail subbands using AQIM.

5.3 Comparative Evaluation

In order to illustrate the advantages of our proposed scheme more clearly, we make a comparison between ours and the scheme proposed by Gomez-Ricardez and Garcia-Hernandez in [16]. The scheme in [16] is chosen for comparison based on the following two similarities. First, just like the manner we have done in this study, it employs a channel coder to protect the watermark. Second, this scheme is also claimed to be robust against the content replacement attack if the affecting portion is less than 20% of the whole audio. Table 3 summarizes the comparison. The method in [16] is indeed capable of restoring the substituted segment when the size of substitution remains unchanged. Restoring the audio segment destroyed by the insertion attack is also possible if the tampered area is accurately located and the whole audio is properly trimmed and aligned. However, dealing with the deletion attack is problematic. For example, deleting a small section of the audio in the middle, then shifting the remaining part ahead, and finally padding zeros at the end can easily cripple the watermark extraction for the scheme in [16]. The cause is ascribable to the fact that the deletion misplaces a large portion of the watermarked audio and thus devastates the channel code information. For the same sake, the scheme in [16] cannot survive the cropping or time-shifting attacks, which are known to disrupt the frame synchronization for correct watermark extraction. By contrast, with the incorporation of the self-synchronization feature discussed in Section 2, the proposed scheme can withstand all the aforementioned attacks (i.e., insertion, deletion, substitution, cropping, and time-shifting).

Table 3 Comparison results

Attack types	**Resistance**	
	The proposed	Scheme in [16]
Cropping / Time shifting	Yes	No
Deletion	Yes	No
Substitution	Yes	Partially feasible
Insertion	Yes	Partially feasible
LSB erasure	Yes	No

Another advantage of the proposed scheme is that it is quite capable of resisting against minor attacks such as LSB erasure. Table 4 presents the extraction results when 1 and 2 LSBs are deliberately obliterated. The results indicate that even in the case of 2 LSBs erasure the proposed scheme can perfectly extract 19 out of 24 embedded watermarks from the 2^{nd} detail subband. Moreover, because the maximum BER (i.e., 1.172%) is less than 20% ×1/8, the Reed-Solomon code capable of correcting 20% erroneous 8-bit symbols is sufficient to recover the original watermark bits.

Table 4 Comparison results

# of erasure bits	**Erase 2 LSBs**		**Erase 1 LSB**	
Embedding location	2^{nd} detail subband	3^{rd} detail subband	2^{nd} detail subband	3^{rd} detail subband
# of watermarks without errors	19	21	22	23
# of watermarks with errors	5	3	2	1
Largest BER among the watermarks with errors	1.172%	0.081%	0.223%	0.001%

6. Conclusion

In this paper, we have proposed a novel watermarking scheme to not only authenticate the veracity and integrity of the received audio but enable the recovery of tampered contents via the exploitation of source-channel coding. After the application of 3-level LWT to the audio signal, the proposed scheme performed two types of watermarking processes in a frame-synchronous manner. A compressed version of the original signal protected with the RS code was embedded into the 3^{rd} and 2^{nd}-level detail subbands using 2^N-ary AQIM, while the frame-related data and hash bits were embedded into the 3^{rd}-level approximation subband using RDM. The experiment results indicated that the watermark embedding resulted in an average SNR of 27.36 dB and an average ODG score around -0.44 for a test set of twenty-four audio clips, suggesting that the

watermarked audio is nearly perceptually indistinguishable from the original one. In the phase of watermarking extraction, the RDM proved to be effective in tracking synchronization codes, thus facilitating the frame alignment and watermark extraction. The 2^N-ary AQIM also demonstrated its competence in performing multi-bit data hiding in the LWT domain. More importantly, the ability of tracing frame boundaries empowered the proposed scheme to combat with the cropping and replacement attacks that no previous self-recovery watermarking schemes could easily handle. As there is plenty of room for hiding extra information in the 3rd-level approximation subband, our future work will be focused on adding other robust watermarks to reinforce copyright protection.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This research work was supported by the Ministry of Science and Technology (MOST), Taiwan, under grant 107-2221-E-197-021.

References

[1] N. Cvejic and T. Seppänen, Digital audio watermarking techniques and technologies: applications and benchmarks. Hershey: Information Science Reference, IGI Global, 2008.
[2] X. He, Watermarking in audio: key techniques and technologies, Youngstown, N.Y.: Cambria Press, 2008.
[3] M. Steinebach and J. Dittmann, "Watermarking-Based Digital Audio Data Authentication," EURASIP Journal on Advances in Signal Processing, vol. 2003, no. 10, pp. 1001-1015, 2003.
[4] M. Q. Fan, P. P. Liu, H. X. Wang, and H. J. Li, "A semi-fragile watermarking scheme for authenticating audio signal based on dual-tree complex wavelet transform and discrete cosine transform," International Journal of Computer Mathematics, vol. 90, no. 12, pp. 2588-2602, 2013.
[5] Ghobadi, A. Boroujerdizadeh, A. H. Yaribakht, and R. Karimi, "Blind audio watermarking for tamper detection based on LSB," Proc. 2013 15th International Conference on Advanced Communications Technology (ICACT), IEEE Press, January 2013, pp. 1077-1082.
[6] N. N. Hurrah, S. A. Parah, N. A. Loan, J. A. Sheikh, M. Elhoseny, and K. Muhammad, "Dual watermarking framework for privacy protection and content authentication of multimedia," Future Generation Computer Systems, vol. 94, pp. 654-673, 2019.
[7] H. He, F. Chen, H. Tai, T. Kalker, and J. Zhang, "Performance analysis of a block-neighborhood-based self-recovery fragile watermarking scheme," IEEE Transactions on Information Forensics and Security, vol. 7, no.1, pp. 185-196, 2011.
[8] Q. Han, L. Han, E. Wang, and J. Yang, "Dual Watermarking for Image Tamper Detection and Self-Recovery," 2013 9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, October 2013, pp. 33-36.
[9] X. Zhang, Z. Qian, Y. Ren, and G. Feng, "Watermarking with flexible self-recovery quality based on compressive sensing and compositive reconstruction," IEEE Transactions on Information Forensics and Security, vol. 6, no. 4, pp. 1223-1232, 2011.
[10] W. L. Tai and Z. J. Liao, "Image self-recovery with watermark self-embedding," Signal Processing: Image Communication, vol. 65, pp. 11-25, July 2018.
[11] S. Sarreshtedari, M. A. Akhaee, and A. Abbasfar, "A watermarking method for digital speech self-recovery," IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP), vol. 23, no.11, pp. 1917-1925, 2015.
[12] W. Lu, Z. Chen, L. Li, X. Cao, J. Wei, N. Xiong, et al., "Watermarking Based on Compressive Sensing for Digital Speech Detection and Recovery (†)," Sensors, vol. 18, no. 7, pp. 2390, 2018.
[13] S. Li, Z. Song, W. Lu, D. Sun, and J. Wei, "Parameterization of LSB in Self-Recovery Speech Watermarking Framework in Big Data Mining," Security and Communication Networks, 2017.
[14] F. Chen, H. He, and H. Wang, "A fragile watermarking scheme for audio detection and recovery," 2008 Congress on Image and Signal Processing, vol. 5, pp. 135-138, 2008.

[15] Menendez-Ortiz, C. Feregrino-Uribe, J. J. Garcia-Hernandez, and Z. J. Guzman-Zavaleta, "Self-recovery scheme for audio restoration after a content replacement attack," Multimedia Tools and Applications, vol. 76, no. 12, pp. 14197-14224, June 2017.
[16] J. J. Gomez-Ricardez and J. J. Garcia-Hernandez, "An audio self-recovery scheme that is robust to discordant size content replacement attack," 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), 2018, pp. 825-828.
[17] Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," IEEE Trans. Information Theory, vol. 47, no. 4, pp. 1423-1443, 2001.
[18] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," Applied and computational harmonic analysis, vol. 3, no. 2, pp. 186-200, 1996.
[19] Daubechies, Ten lectures on wavelets. Philadelphia, 1992.
[20] H. T. Hu and L. Y. Hsu, "A DWT-based rational dither modulation scheme for effective blind audio watermarking," Circuits, Systems, and Signal Processing, vol. 35, no. 2, pp. 553-572, 2016.
[21] H. T. Hu and L. Y. Hsu, "Supplementary schemes to enhance the performance of DWT-RDM-based blind audio watermarking," Circuits, Systems, and Signal Processing, vol. 36, no. 5, pp. 1890-1911, 2017.
[22] X. He and M. S. Scordilis, "An enhanced psychoacoustic model based on the discrete wavelet packet transform," Journal of the Franklin Institute, vol. 343, no. 7, pp. 738-755, 2006.
[23] T. Painter and A. Spanias, "Perceptual coding of digital audio," Proc. IEEE, vol. 88, no. 4, pp. 451-515, 2000.
[24] H. Traunmüller, "Analytical expressions for the tonotopic sensory scale," The Journal of the Acoustical Society of America, vol. 88, no. 1, pp. 97-100, 1990.
[25] H. T. Hu, L. Y. Hsu, and H. H. Chou, "Variable-dimensional vector modulation for perceptual-based DWT blind audio watermarking with adjustable payload capacity," Digital Signal Processing, vol. 31, pp. 115-123, 2014.
[26] H. T. Hu and L. Y. Hsu, "Robust, transparent and high-capacity audio watermarking in DCT domain," Signal Processing, vol. 109, pp. 226-235, 2015.
[27] H. Hu and T. Lee, "High-Performance Self-Synchronous Blind Audio Watermarking in a Unified FFT Framework," IEEE Access, vol. 7, pp. 19063-19076, 2019.
[28] S. Lin and D. J. Costello, Error Control Coding, Second Edition: Prentice-Hall, Inc., 2004.
[29] G. Forney, Jr., "On decoding BCH codes," IEEE Trans. Information Theory, vol. 11, no. 4, pp. 549-557, 1965.
[30] B. den Boer and A. Bosselaers, "Collisions for the compression function of MD5," Workshop on the Theory and Application of Cyptographic, Berlin, Heidelberg, pp. 293-304, 1994.
[31] ITU-R Recommendation BS.1387, "Method for objective measurements of perceived audio quality," December 1998.
[32] P. Kabal, "An examination and interpretation of ITU-R BS.1387: Perceptual evaluation of audio quality," TSP Lab Technical Report, Dept. Electrical & Computer Engineering, McGill University, 2002.