A novel audio watermarking in wavelet domain[*]

Ming Li, Yun Lei, Jian Liu and Yonghong Yan

ThinkIT lab, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100080, P. R. China

{mli2,ylei,jliu,yyan}@hccl.ioa.ac.cn

Abstract

A novel approach is proposed for robust audio watermarking in wavelet domain. It emphasizes on enhancing security by dynamically modifying embedding strategy. The modification is based on real-time changes of the watermark information and host audio. Without using any secret key, informative watermark are embedded into audio with different strength. The distribution of watermarked audio in wavelet domain is relatively similar to clean audio, compared to existing watermarking schemes in wavelet domain. The analysis and experimental results are given to demonstrate that the proposed watermarking scheme is effective against common signal processing manipulations and attacks, such as Gaussian noise corruption, resampling, requantization, MP3 compression and D/A-A/D conversion.

1. Introduction

The rapid growth of the broadband communication networks and proliferation of audio distribution have raised the issue of intellectual property rights protection. Audio watermarking, as a powerful technique for copyright protection for audio works, has been widely exploited [13, 4]. A digital watermarking is imperceptibly and robustly embedded into the host data such that it cannot be removed [4]. The audio watermark is imperceptible to humans and is embedded into the cover audio media. The vulnerability to random shifting, cropping, and time scaling attacks is one of the main weaknesses of many existing watermarking algorithms [1, 6]. Embedding the synchronization codes into the original audio together with the informative watermark is an effective method to withstand random shifting, cropping, and time scaling attacks [14, 5]. By detecting the synchronization codes, the position where the informative

watermark is embedded can be located. If the synchronization codes are hidden in time domain [5, 8], they are not adequately robust due to the limited embedding strength and audio signal instability in time domain. On the other hand, if the synchronization codes are embedded into frequency domain, the computational complexity to search the synchronization codes grows explosively, although the synchronization codes become more robust. The synchronization codes based on discrete wavelet transformation (DWT) are a reasonable tradeoff between robustness and computational cost [14]. In general, watermarks are embedded into the low frequency coefficients in DWT domain to achieve robust performance against common signal processing procedures and noise corruptions [14, 3]. In this paper, a novel audio watermarking method in wavelet domain is proposed. The method is designed to enhance security by the predetermined embedding strengths and kept similar distribution to that of the corresponding clean audio in wavelet domain. Before the watermark extraction, embedding strategy is calculated according to the received audio. Because both embedding and extracting strategies are dynamically generated for different audio and embedded information. Thus the different embedding strengths are created for each audio, the same watermarks are embedded differently from audio to audio. Hence security of the algorithm is ensured. This paper is organized as following. In Sect.2 watermark encoding and decoding algorithms are explained in detail. The performance of the proposed algorithm is evaluated in Sect.3. Experimental results are given in Sect.4 in terms of detection accuracy for a wide assortment of signal processing operations and distortions. The concluding remarks are presented in Sect.5.

2. The Proposed Algorithm

Overview of the proposed watermark embedding and extracting algorithm is depicted below. First the input original audio signal is segmented into many sections, and these sections are transformed to wavelet domain by 5-level wavelet decomposition. Then a sequence $\{m_i\}$ consisted of synchronization codes and informative watermark bits [14] is

---

embedded into the low frequency sub-band DWT coefficients of the corresponding section by a secure efficient method. Finally, the inverse discrete wavelet transformation (IDWT) is applied to the modified wavelet coefficients to transform them back into the time domain.

2.1. Data Embedding

A watermarking method in the category of quantization index modulation [2] is used to embed each bit of a binary sequence $\{m_i\}$ of which the elements are between two alternative values (i.e. 1, -1.) by modifying the low frequency wavelet coefficients with a block of even length K. We divide each section into L blocks with the length of K, and each bit of $\{m_i\}$ is embedded to the corresponding block. So the length of each audio section is determined by the amount of informative watermark, the number of DWT decomposition levels, and the length of each block. The detail procedure for embedding one section is as follows.

1. We first split the original composite audio data into proper sections; using "Haar" filter [14] and decomposition depth of five steps, we perform 5-level DWT on every section. For each section, let N and N_w denote the numbers of original data in time domain and low-frequency sub-band coefficients in wavelet domain respectively and $N_w = N/2^5$. Assuming $L = L_s + L_i$, where L_s and L_i denote the length of synchronization codes and informative watermark bits in each section respectively, the relationship among L, N_w and K is $L = \frac{N_w}{K}$.

2. To automatically adjust the total embedding strength of each section by the alternating current (AC) energy, the sample mean is calculated as (1):

$$\overline{W} = \frac{1}{N_w} \sum_{i=1}^{N_w} W_i \qquad (1)$$

Where $W = \{W_1, \ldots, W_{N_w}\}$ are low-frequency sub-band coefficients whose index denote placement inside the low sub-band. And we define $W^{I_i} = \{W_1^{I_i}, \ldots, W_k^{I_i}\}$ as the sequence including K elements in the I_ith block, correspondent to the index of $I = \{I_1, \ldots, I_L\}$. Using the random index makes it more secure to locate and extract synchronization codes. The AC energy is calculated as (2):
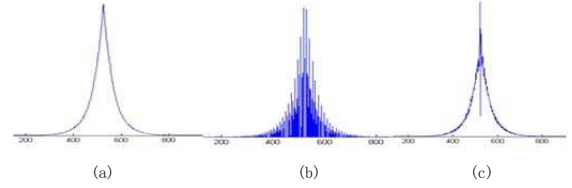
$$E_s = \sum_{i=1}^{N_w} (W_i - \overline{W})^2 \qquad (2)$$

Then the total embedding strength in one section is generated as (3):

$$\sum_{i=1}^{L} S_i = \sqrt{E_s} \cdot S_0 / S_T \qquad (3)$$

Table 1. The fluctuation of $\sqrt{E_s}$ in common signal processing manipulations.

Attacks	Max(%)	Min(%)	Average(%)
$GNC(15dB)$	1.01	5.25×10^{-7}	9.98×10^{-2}
$Resampling(6kHz)$	1.90	8.54×10^{-6}	6.08×10^{-2}
$Requantization(8Bit)$	8.45	1.55×10^{-6}	8.04×10^{-2}
$MP3(32kbps)$	4.15	3.85×10^{-4}	0.37
$D/A - A/D conversion$	9.27	8.27×10^{-5}	0.31

(a)　　　　(b)　　　　(c)

Figure 1. Histogram of wavelet coefficient in clean audio, referenced Wu's method, and proposed method. (a) Histogram of wavelet coefficient in clean audio; (b) Histogram of wavelet coefficient in Wu's method (steady embedding strength); (c) Histogram of wavelet coefficient in proposed method (self-adjusting embedding strength).

Where S_0 denotes the referenced embedding strength and S_T denotes a parameter. They are both constants during the whole watermarking process. S_i denotes the embedding strength in each section so $\sum_{i=1}^{L} S_i$ is based on the magnitude of E_s, which in general is approximately invariable.

The reason that the AC energy is chosen to adjust the total embedding strength automatically is that the energy parameter generally is an approximate constant in common signal processing manipulations and attacks. In our experiment, we test the property in Gaussian Noise Corruption (GNC), resampling, requantization, MP3 compression and D/A-A/D conversion. 2500 16-bit signed mono audio signals sampled at 44.1 kHz with the length of about one second (the length of one section, 45056 samples) in WAVE format are employed for these statistics. The result is present in Table 1.

The strategy that the total embedding strength is adjusted by the AC energy is efficient to improve the security against steganalysis. Because the histogram is widely used for steganalysis [12, 11], we show the DWT histogram contrast between steady embed-

ding strength in Wu's method which use the same embedding strength for all the modifications of low frequency wavelet coefficients [14] and self-adjusting embedding strength employed by proposed method in Fig.1. It presents that wavelet coefficient distribution of the self-adjusting embedding strength method is more similar with that of clean audio than steady embedding strength method. The difference between Fig.1a and Fig.1c is avoidable by keeping clean in the smaller wavelet coefficients. So steganalysis using histogram differences become more difficult and less efficient to the self-adjusting embedding strength method.

3. As mentioned above, there are L blocks in each section, and in each block there are K low frequency sub-band wavelet coefficients.

Define $A^i = \{a_1^i, \ldots, a_{K/2}^i\}$ as the subset of W whose subscripts and superscripts correspond to the first $K/2$ elements of the index set and the elements are in the ith block respectively with a similar definition for $B^i = \{a_{k/2+1}^i, \ldots, a_K^i\}$ with the last $K/2$ elements. There are $A^i = \{W_1^{I_i}, \ldots, W_{K/2}^{I_i}\}$ and $B^i = \{W_{K/2+1}^{I_i}, \ldots, W_K^{I_i}\}$, for $i = 1, \ldots, L$. For avoiding the influence from direct current (DC) in signal processing, we calculate the variable as the carrier to embed the watermark bit by (4).

$$C_i = \sum_{j=1}^{k/2} W_j^{I_i} - \sum_{j=k/2+1}^{k} W_j^{I_i} = \sum_{j=1}^{k/2} A_j^i - \sum_{j=1}^{k/2} B_j^i \quad (4)$$

4. In order to enhance security, the embedding strength for each section $\{S_i\}$ is dynamically modified with real-time changes of the watermarks and host audio so the changes of $\{S_i\}$ must be automatically detected from the watermarked signal in the data extraction step. In order to correctly obtain $\{S_i\}$ in extraction, $\{S_i\}$ calculated from the watermarked signal have to be the same as the one calculated from the original signal.

The algorithm allows $\{S_i\}$ of each section to be automatically adjusted during the iteration by the constant total embedding strength $\sum_{i=1}^{L} S_i$ and the scale of each absolute carrier value of $\{C_i\}$. Specifically, the rule for embedding is a watermarking technique in the category of quantization index modulation as follows:

We first define the following parameters:

1) Total embedding strength of this section: $G = \sum_{i=1}^{L} S_i$;

2) Average embedding strength: $\overline{S} = G/L = \sum_{i=1}^{L} S_i/L$;

3) A threshold value for convergence: ξ_0;

4) The nth iteration result of S_i, Y_i, C_i: $S_i^{(n)}$, $Y_i^{(n)}$, $C_i^{(n)}$;

5) The corresponding embedded carrier: $\{C_i'\}$.

The algorithm:

Step 1. Arbitrarily initialize $Y_i^{(0)}$ as a uniform array.

Step 2. Update $\{S_i\}$ and $\{Y_i\}$ by the flow as follows (5,6,7):

$$S_i^{(n)} = \begin{cases} \overline{S} & \text{if } T \geq \overline{S} \\ \overline{S}/2 & \text{if } T \leq \overline{S}/2 \\ T & \overline{S}/2 < T < \overline{S} \end{cases}$$

$$\text{where} \quad T = G \cdot \frac{\frac{1}{(|Y_i^{n-1}|)}}{\sum_{i=1}^{L} \frac{1}{(|Y_i^{n-1}|)}} \quad (5)$$

$$S_i^{(n)} = (S_i^{(n)} + S_i^{(n-1)})/2 \qquad \text{if } (n \geq 2) \quad (6)$$

$$Y_i^{(n)} = \lfloor |C_i^{(n)}|/S_i^{(n)} \rfloor \cdot S_i^{(n)} + S_i^{(n)} \cdot (1/2 + 1/4 \cdot m_i) \quad (7)$$

Step 3. Compute the distance between $Y_i^{(n)}$ and $Y_i^{(n-1)}$:

$$\xi^{(n)} = Y_i^{(n)} - Y_i^{(n-1)} \quad (8)$$

Step 4. If $\xi^{(n)} > \xi_0$, the iteration is not convergent, then go to Step 2; else if $\xi^{(n)} < \xi_0$, the iteration is convergent, go to Step 5.

Step 5. Terminate the iteration, and $\{S_i^{(n+1)}\}$ is as the same as $\{S_i^{(n)}\}$. In this case, the embedding strength $\{S_i\}$ calculated at the data-extracting step is equal to the ones in encoding regardless of noise.

Step 6. Let $\{Y_i\}$ be the new $\{C_i'\}$.

5. After modulation of the variable $\{C_i\}$, we have L elements $\{C_i'\}$, and the difference between C_i and C_i' should be uniformly distributed among the K low frequency sub-band wavelet coefficients in the specified I_ith block. However, there will be some signal discontinuities after amplitude scaling near section boundaries which will then cause "click" sounds that are perceivable to human ears. For smoothing and weakening the break between two adjoining blocks, the wavelet coefficients' modification near the boundary should be smoothed [9]. We use the following sequence R with the length of $K/2$ to modulate their magnitude when the block size K is relatively minor, and more elaborate smooth sequences are employed for larger blocks.

$$\begin{aligned} R &= \{R_j, j = 1, \ldots, k/2\} \\ &= \{1/5, 3/5, 1, \ldots, 1, 3/5, 1/5\} \quad (9) \end{aligned}$$
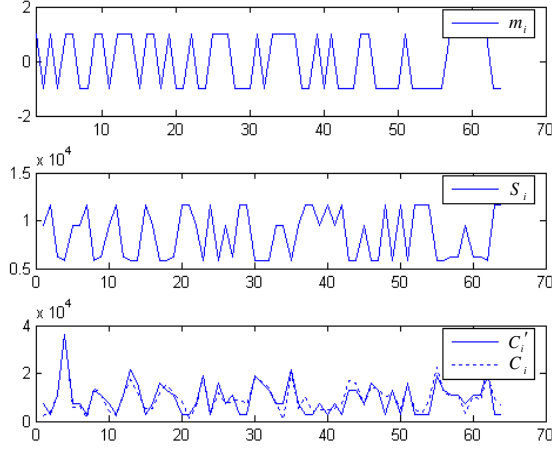
Figure 2. An example about relationship among S_i, m_i, C_i and C_i' is demonstrated.

Specifically, the rule for modifying low frequency sub-band wavelet coefficients is defined as follows

$$\{(A_j^i)'\} = \{A_j^i\} + \frac{1}{2}(C_i' - C_i) \cdot (R_j / \sum_{i=1}^{K/2} R_i)$$
$$j = 1, \ldots, K/2$$
$$\{(B_j^i)'\} = \{B_j^i\} - \frac{1}{2}(C_i' - C_i) \cdot (R_j / \sum_{i=1}^{K/2} R_i)$$
$$j = 1, \ldots, K/2$$
$$(10)$$

So the whole watermarked low frequency sub-band wavelet coefficients $\{W_i'\}$ in one section are the combination of $\{(A_j^i)'\}$ and $\{(B_j^i)'\}$.

6. IDWT is applied to $\{W_i'\}$ to transform it back to the time-domain signal. The change of the embedding strength by the iteration makes the watermark embedded in different sections with different strengths, even if the total energy of each section is the same with others. The embedding method depends on the watermark information and carrier content, rather than any key words.

2.2 Data Extraction

For data extraction in each section, according to (1,2,3) mentioned in Sect.2, low-frequency sub-band coefficients $W = \{W_1, \ldots, W_{N_w}\}$, total embedding strength $G = \sum_{i=1}^{L} S_i$, average embedding strength $\overline{S}$, and the carrier variable $\{C_i\}$ are calculated to extract the watermark bits $\{M_i\}$. Then calculate S_i for each block.

$$S_i^{(n)} = \begin{cases} \overline{S} & \text{if } T \geq \overline{S} \\ \overline{S}/2 & \text{if } T \leq \overline{S}/2 \\ T & \overline{S}/2 < T < \overline{S} \end{cases}$$

$$\text{where} \quad T = G \cdot \frac{\frac{1}{(|C_i^{n-1}|)}}{\sum_{i=1}^{L} \frac{1}{(|C_i^{n-1}|)}} \quad (11)$$

At the next step, we calculate binary watermark bits as follows [14, 10]:

$$M_i = \left\{ \begin{array}{ll} 1 & \text{if } |C_i| - \lfloor |C_i|/S_i \rfloor \cdot S_i \geq S_i/2 \\ 0 & \text{if } |C_i| - \lfloor |C_i|/S_i \rfloor \cdot S_i < S_i/2 \end{array} \right\}$$
$$i = 1, \ldots, L \quad (12)$$

3 Performance Analysis

In this section, we evaluate the performance of the proposed algorithm in terms of SNR with referenced embedding strength S_0 and the stability of AC energy in watermark-embedding processing.

3.1 Embedding strength analysis

Based on descriptions in Sect.2, if the total embedding strength in one section is generated by (3), the average embedding strength $\overline{S_w}$ of each wavelet coefficient, regardless of the processing of the iteration-based dynamic embedding strengths and average embedding strength modulation where the average embedding strength is limited between $\overline{S}$ and $\frac{\overline{S}}{2}$, can be expressed as:

$$\overline{S_w} = \frac{\overline{R}}{K \cdot L} \sqrt{E_s} \cdot S_0 / S_T \quad (13)$$

where $\overline{R}$ notes the offset generated by window R:

$$\overline{R} = \sqrt{\frac{1}{K/2} \sum_{j=1}^{K/2} \left(\frac{K}{2} \cdot \frac{R_j}{\sum_{i=1}^{K/2} R_i} \right)^2} \quad (14)$$

And then from Wu's analysis, the difference of a section in wavelet domain between clean and watermarked audio is:

$$Exp.((W_i - W_i')^2) = \frac{7\overline{S_w}^2 KL}{48} = \frac{7\overline{R}^2}{48KL} E_s \cdot S_0^2 / S_T^2 \quad (15)$$

Where $Exp.$ stands for expectation. If the energy of clean original audio in one section is E, the SNR between the original audio and the watermarked audio is generated as follows:

$$SNR = -10 log_{10} \left(\frac{7\overline{R}^2}{48KLS_T^2} \cdot \frac{E_s}{E} \cdot S_0^2 \right) \quad (16)$$

In fact, the processing of the iteration-based dynamic embedding strengths and average embedding strength modulation could change the distribution of the embedding

Table 2. The fluctuation of $\sqrt{E_s}$ by watermark embedding processing.

$SNR(dB)$	Max(%)	Min(%)	Average(%)
20	10.18	8.29×10^{-5}	0.39
25	2.82	7.73×10^{-5}	0.13
30	0.63	1.60×10^{-5}	5.09×10^{-2}
35	0.27	3.31×10^{-5}	2.68×10^{-2}
40	0.17	6.38×10^{-6}	1.43×10^{-2}

strengths in one section to generate the experiential SNR's offset δ_{SNR}. The modified SNR equation is:

$$SNR = -10log_{10}(\frac{7\overline{R}^2}{48KLS_T^2} \cdot \frac{E_s}{E} \cdot S_0^2) + \delta_{SNR} \quad (17)$$

The SNR could be estimated by the referenced embedding strength S_0 by the above equation. By the statistical experience, the offset δ_{SNR} is 2.55 dB.

### 3.2 The stability of AC energy in watermark-embedding processing

In Sect.2, that the AC energy is a constant in common signal processing manipulations is confirmed, but watermark-embedding processing also modifies wavelet coefficients. In this section, we show that the fluctuation of the AC energy after watermarks are embedded into carriers in different embedding strength in Table 2, from which the average fluctuation of AC energy after imperceptibly embedding watermark is tiny so that the fluctuation has little disturbance in decoding.

4 Experimental Result

In our experiment, six 16-bit signed mono audio signals with different proprieties: piano, rock, electric guitar, organ, saxophone, and pop, sampled at 44.1 kHz with the length

Table 3. Subjective Diff-Grades

SDG	Description
0.0	Imperceptible
-1.0	Perceptible, but not annoying
-2.0	Slightly annoying
-3.0	Annoying
-4.0	Very annoying
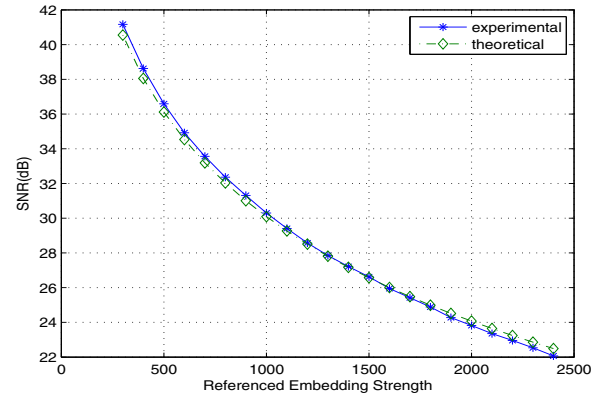
Figure 3. SNR **and Referenced Embedding Strength** S_0 **for** $pop.wav$**.**

of about 112 seconds in WAVE format are employed. The length of watermark bits L in one section is 64, the number of the elements in one block K is 22, the constant S_T is 213 and the threshold value for convergence ξ_0 is 0.001. The values of referenced embedding strength are 800 for piano, 1500 for rock, 2000 for electric guitar (E-guitar), 1500 for organ, 1000 for saxophone, and 2600 for pop respectively. The SNRs between the original audio and the watermarked audio are 30.64dB for piano, 24.15dB for rock, 22.12dB for E-guitar, 22.26dB for organ, 28.91dB for saxophone, and 21.68dB for pop. Objectively evaluating audio quality with respect to the original signal is difficult, for instance amplitude-scaled audio that may be able to maintain good perceptual quality, but at the expense of large mean square errors [9]. Therefore, a series of subjective tests by ten listeners using subjective diff-grades (SDG) were performed. The meaning of each score of SDG is shown in Table 3 and the evaluation results of subjective listening test and robust experiment are presented together in Table 4.

It is apparent that all the average SDG scores are zero or very close to zero which means that the watermarked audio and the original audio are perceptually undistinguishable. In Gaussian noise corruption 1, the signal-to-noise ratios (SNRs) are 13.49dB for piano, 13.55dB for rock, 13.19dB for E-guitar, 13.97dB for organ, 13.46dB for saxophone, and 13.92dB for pop. In Gaussian noise corruption 2, the SNRs are 19.51dB for piano, 19.57dB for rock, 19.21dB for E-guitar, 19.99dB for organ, 19.49dB for saxophone, and 19.94dB for pop. The D/A-A/D conversion referred to the D/A-A/D attack model presented in [15] and the used sound card in my experiment is SoundBlast 5.1 live in Creative. Through Table4, we can see that the proposed method is robust against various attacks. As mentioned in [15],

Table 4. Average SDG score of subjective listening test and the BER(%) of manipulations.

	Piano	Rock	E-guitar	Organ	Saxophone	Pop
Gaussian noise corruption1	8.78	0.19	0.02	0.58	6.05	0.02
Gaussian noise corruption2	2.31	0.06	0.00	0.02	1.47	0.00
Resampling (6000Hz)	0.00	0.11	0.02	0.00	0.00	0.06
Requantization (8bits)	0.80	0.08	0.00	0.00	0.05	0.00
MP3 (128kbps)	0.00	0.00	0.00	0.00	0.00	0.00
MP3 (56kbps)	0.00	0.39	0.14	0.00	0.00	0.02
MP3 (32kbps)	0.08	1.95	0.25	0.25	0.00	0.30
D/A-A/D conversion	0.00	0.31	0.03	3.08	0.28	0.78
Average score of SDG	-0.2	-0.1	0	-0.2	-0.1	0

quantization-based audio watermarking is very susceptible to D/A-A/D conversion due to the modification of signal energy, phase changes, and noises corruption. However, the proposed method employs AC energy to overcome the modification of signal energy, uses the combination of coefficients in a whole block to reduce the affection of noise corruption and adopts synchronization scheme against phase changes so called temporal scaling in [15] which guarantee the robustness to D/A-A/D conversion. The different bit-number distribution between synchronization codes and informative watermark results in different information capabilities and synchronization rates. Figure 3 shows that the experimental and theoretical results of SNR between the origin audio and the watermarked audio are a close match.

5 Conclusions

A novel audio watermarking method in wavelet domain was presented. The watermarking scheme consists of quantization modulation, and the dynamic changes of the embedding strength based on the contents. We claim that enough changes of the embedding strength are provided to make sure that there do not exist the same embedding strength between different audios. From the experiment results, it can be concluded that the proposed approach can provide adequate robustness against various vicious detections or extractions.

References

[1] P. Bassia, I. Pitas, and N. Nikolaidis. Robust audio watermarking in the time domain. *IEEE Trans. Multimedia*, 3(2):232–241, June 2001.

[2] B. Chen and G. W. Wornell. Quantization index modulation methods for digital watermarking and information embedding of multimedia. *Journal of VLSI Signal Processing*, 27:7–33, 2001.

[3] N. Cvejic and T. Seppanen. Robust audio watermarking in wavelet domain using frequency hopping and patchwork method. *ISPA*, 1:251–255, 2003.

[4] F. Hartung and M. Kutter. Multimedia data-embedding and watermarking technologies. *Proceedings of the IEEE*, 87(7):1079–1107, July 1999.

[5] J. Huang, Y. Wang, and Y. Q. Shi. A blind audio watermarking algorithm with self-synchronization. *ISCAS*, 3:627–630, 2002.

[6] B. Ko, R. Nishimura, and Y. Suzuki. Time-spread echo method for digital audio watermarking using pn sequences. *ICASSP*, 2:2001–2004, 2002.

[7] W. Li, X. Xue, and P. Lu. Localized audio watermarking technique robust against time-scale modification. *IEEE Trans. Multimedia*, 8(1):60–69, February 2006.

[8] W. Lie and L. Chang. Robust and high-quality time-domain audio watermarking subject to psychoacoustic masking. *ISCAS*, 2:45–48, 2001.

[9] W. Lie and L. Chang. Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modification. *IEEE Trans. Multimedia*, 8(1):46–59, February 2006.

[10] J. Liu. The application of wavelet in image compression and digital watermarking. *Zhejiang University doctoral dissertation*, 3, 2001.

[11] S. Liu, H. Yao, and W. Gao. Steganalysis of data hiding techniques in wavelet domain. *Proceedings of International Conference on Information Technology: Coding and Computing Proceedings*, 1:751–754, 2004.

[12] M. Niimi, R. O. Eason, H. Noda, and E. Kawaguchi. Intensity histogram steganalysis in bpcs steganography. *Proceedings of SPIE: Security and Watermarking of Multimedia Contents III*, 4314:555–564, 2001.

[13] M. D. Swanson, M. Kobayashi, and A. H. Tewfik. Multimedia data-embedding and watermarking technologies. *Proceedings of the IEEE*, 86(6):1064–1087, June 1998.

[14] S. Wu, J. Huang, D. Huang, and Y. Q. Shi. Efficiently self-synchronized audio watermarking for assured audio data transmission. *IEEE Trans. Broadcasting*, 51(1):69–76, March 2005.

[15] S. Xiang and J. Huang. Analysis of d/a and a/d conversions in quantization-based audio watermarking. *International Journal of Network Security*, 3(3):230–238, November 2006.

COMPUTER SOCIETY