IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 30, 2018, accepted January 8, 2019, date of publication January 18, 2019, date of current version February 22, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2893646*

# High-Performance Self-Synchronous Blind Audio Watermarking in a Unified FFT Framework

**HWAI-TSU HU, (Member, IEEE), AND TUNG-TSUN LEE**
Department of Electronic Engineering, National I-Lan University, I-Lan 26047, Taiwan

Corresponding author: Hwai-Tsu Hu (hthu@niu.edu.tw)

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

**ABSTRACT** This paper presents a blind audio watermarking method that uses two different schemes to hide binary bits and auxiliary information within separate ranges of a fast Fourier transform (FFT) sequence. An adaptive vector norm modulation (AVNM) scheme is introduced to achieve a satisfactory balance of imperceptibility, robustness, and payload capacity. An improved spread spectrum (ISS) scheme is developed to produce a striking correlation peak, which facilitates the detection of synchronization codes in the FFT domain. The combination of robust audio segment extraction and recursive FFT makes it possible to execute these two FFT-based schemes in tandem on a sample-by-sample basis. The experiment results confirm that watermark embedding causes merely a negligible degradation in perceptual quality. A detectability test proved the effectiveness of the ISS scheme in self-synchronization as well as hiding auxiliary data. Three versions of AVNM with capacities ranging from 344.53 to 1033.59 bits per second were demonstrated. Compared with six recently developed schemes, AVNM exhibited advantages in terms of negligible quality distortion, flexible payload capacity, and excellent robustness against a variety of common signal processing attacks.

**INDEX TERMS** Synchronous blind audio watermarking, fast Fourier transform, adaptive vector norm modulation, improved spread spectrum, robust audio segment extractor.

## I. INTRODUCTION

The ease with which multimedia data can be reproduced, modified, and distributed makes it very easy to infringe on intellectual property rights. Digital watermarking is one of the most effective approaches to protect copyrighted materials. Specifically, digital watermarks (e.g., logos) embedded in noise-tolerant signals, such as audio, video, or image files, can later be extracted to prove ownership and/or authenticate content [1], [2].

Depending on the application scenario, watermarks can be classified as fragile or robust. Robust watermarks are meant to be resilient to modification attempts, whereas fragile watermarks are used to detect changes in multimedia data without compromising the fidelity of the original signal. The three primary concerns in robust audio watermarking are imperceptibility, robustness, and capacity [1]–[3]. An ideal watermarking scheme provides sufficient payload capacity to contain all necessary information. Once embedded, the watermark must be inaudible to the human ear and of sufficient robustness to withstand malicious attacks.

Watermarking methods can be divided into non-blind (including semi-blind) and blind methods, based on the information required for watermark recovery. Non-blind methods require the original multimedia source and/or watermark for extraction, whereas blind methods require neither. Investigators have previously explored a variety of representations for blind audio watermarking. Watermarking in the time domain involves a direct adjustment of audio samples in accordance with the characteristics of the watermark. Watermarking in the transform domain is executed on feature coefficients drawn from the host audio signal. Transform domain methods are the most popular, due to their capacity to exploit signal characteristics and/or human auditory properties. Typical transform methods include discrete cosine transform (DCT) [4]–[7], discrete Fourier transform (DFT) [8]–[11], discrete wavelet transform (DWT) [5], [12]–[15], and singular value decomposition (SVD) [16]–[18]. These methods provide good performance in terms of robustness and imperceptibility; however, the computational overhead is usually higher than that of methods implemented in the time domain.

In the past, researchers developed a variety of schemes to enhance the performance of watermarking, including quantization index modulation (QIM) [19], [20], spread spectrum (SS) [21]–[23], echo-hiding [24], [25], and patchwork [6], [26]. Among these schemes, QIM and its variants have been the most popular because they provide a reasonable tradeoff among three conflicting requirements: capacity, imperceptibility, and robustness. The SS-based approach has also attracted considerable attention due to its resistance to tampering and the simple yet efficient processes that it employs. Implementing SS in the transform domain allows human perceptual properties to be exploited and prevents compression-related data loss. Furthermore, most SS-based implementations are blind, which means that watermark detection can be performed in the absence of the original signal.

Audio watermarking methods can also be classified according to the processing strategy that is used during embedding. Some watermarking methods require the entire audio signal to embed the watermark; however, most schemes divide the audio signal into frames which are processed individually due to the computational cost of dealing with a long audio signal. Nonetheless, the strategy of frame partition simply shifts the computational burden to extraction, because the watermarking positions must be identified prior to extraction. Another drawback to performing watermarking on the entire file is a lack of flexibility in terms of payload capacity, as the amount of information to be embedded is fixed, regardless of the length of the audio signal.

When extracting watermarks that have been embedded using frame-oriented schemes, recovering the exact position of each frame is critical. Several techniques had been proposed for synchronization and, as with embedding, synchronization techniques can be implemented in the time domain or in the transform domain. Most of the methods in the transform domain demand intensive computational power due to the repetitive tasks involved in searching for the watermark. For this reason, time-domain methods are typically preferred for real-time applications. However, time-domain methods also tend to be vulnerable to malicious attacks. In light of the aforementioned discussion, we sought to develop a self-synchronous blind audio watermarking scheme that employs an FFT framework. Audio watermarking in the FFT domain may provide additional advantages, as such techniques exploit human auditory properties which are best described in the frequency domain. Moreover, FFT has proven well-suited to high-capacity watermarking [8], [9].

The remainder of this paper is organized as follows. Section II presents two novel schemes which can be used in tandem to attain high-performance self-synchronous blind audio watermarking in the FFT domain: adaptive vector norm modulation (AVNM) and improved SS (ISS). AVNM is a QIM-based scheme that is employed to embed a large quantity of binary bits in low frequency FFT components. ISS not only enables efficient frame synchronization but conveys additional information. Section III details the procedures used in watermark embedding and extraction. Section IV evaluates the proposed schemes in terms of imperceptibility, robustness, and payload capacity. We also present a comparison with six recently developed watermarking schemes. Conclusions are drawn in Section V.

## II. WATERMARKING IN THE FFT DOMAIN

Discrete Fourier transform (DFT) is a process which involves decomposing a signal into a combination of frequency components. Fast Fourier transform (FFT) is a means to compute the same result more quickly. FFTs are of considerable importance in a variety of digital signal processing applications, including audio watermarking.

The proposed watermarking method embeds two types of information in two separate parts of the FFT sequence using two different tactics. The methods presented in [4] and [15] suggest two rules applicable to audio watermarking. First, the use of a larger number of vector coefficients tends to enhance resistance to attacks. As the payload capacity also depends on the number of coefficients gathered in each vector, the capacity decreases whenever the number of coefficients increases. Second, it is preferable to adapt embedding strength to the spectral distribution of the audio signal. One commonly used strategy to resolve the contradictory requirements between imperceptibility and robustness involves increasing the embedding strength to the maximum level that can be achieved without introducing perceptual distortion. The methods in [4], [13], and [15] are typical examples in which the embedding strength is adjusted adaptively to achieve high performance watermarking.

The proposed watermarking method first involves searching for audio segments that would be appropriate for watermark embedding. Audio segments which lack sufficient intensity should be avoided as they are unlikely to render effective watermarks. Each embeddable audio segment is then partitioned into non-overlapping frames of length $L_f$. After applying FFT to the selected audio frame, two frequency ranges are allocated for the embedding of watermark bits and synchronous information. In this study, we adopt a frame length of $2^{13}$; i.e., $L_f = 8192$. As will be clarified in the subsequent discussion, such a frame length renders a necessary amount of FFT coefficients to secure effective watermarking.

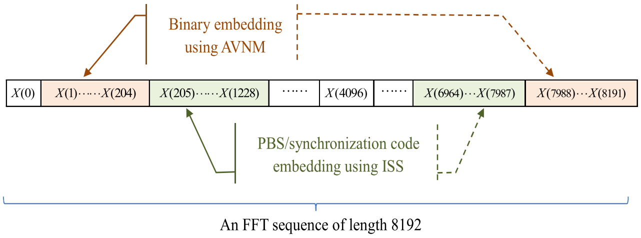


**FIGURE 1. The proposed watermarking schemes that are applied to separate sections of the FFT sequence.**

Figure 1 illustrates the watermarking objects in the FFT domain. In this illustration, binary data are embedded in the low-frequency region. A pseudorandom bipolar sequence (PBS) is then placed in succeeding FFT coefficients. In this study, we deliberately controlled the embedding strength below the masking threshold to achieve adequate imperceptibility. The energy of an audio signal is generally concentrated in the low-frequency region; therefore, these frequency components tend to provide higher tolerance for alterations introduced by the watermark. Accordingly, embedding the watermark in low-frequency FFT coefficients can receive better robustness.

Accurately extracting the embedded watermark requires precise identification of the frame position. The existence of a synchronization mark can be verified by examining the correlation function between the FFT sequence and the synchronization mark. For SS conducted in the FFT domain, this can be excessively time consuming due to the computational power required to perform FFT and correlation function associated with every possible move of frame location. One possible solution is to use a fast algorithm to accelerate the process. Another solution is to employ a locator to perform a coarse demarcation and then refine the frame position afterwards. In this study, we exploited both schemes in order to maximize efficiency.

### A. WATERMARK EMBEDDING USING ADAPTIVE VECTOR NORM MODULATION (AVNM) IN THE FFT DOMAIN

Many watermarking schemes belong to the QIM class because this class of schemes achieves provably good rate-distortion-robustness performance [19]. In QIM watermarking, the embedding strength is characterized by the choice of the quantization step size. A large step size is conducive to robustness but detrimental to audio quality. In contrast, a smaller step size is conducive to imperceptibility but tends to compromise robustness. Selecting the appropriate quantization step size involves seeking a reasonable tradeoff among imperceptibility, data payload, and robustness. In [4], [13], and [15], it was demonstrated that the quantization step size can be adaptively retrieved from the watermarked audio as long as the energy level remains unchanged throughout the watermarking process.

Based on the same principle, we introduce an adaptive scheme to perform binary watermarking in the FFT domain. For this, once an audio frame of size $L_f$ is converted into an FFT sequence (termed $\{X(k)|k = 0, 1, \cdots, L_f - 1\}$), we divide the leading $L_X$ coefficients (d.c. term excluded) into $L_\rho$ subgroups of size $q$ coefficients. Without a loss of generality, it is assumed here that $L_X(192)$ is divisible by $q$. The magnitude FFT coefficients in each subgroup form a vector:

$$\begin{aligned}\mathbf{x}_m = \left[\left|X(k_{m,1})\right| \left|X(k_{m,2})\right| \cdots \left|X(k_{m,q})\right|\right]^T;\\ k_{m,n} \in \{1, 2, \cdots, L_X\} \quad m = 0, 1, \cdots, L_\rho - 1;\\ n = 0, 1, \cdots, q-1;\ L_\rho = L_X/q = 192/q. \qquad (1)\end{aligned}$$

where $|\cdot|$ denotes the complex magnitude. $k_{m,n}$ represents the $n^{th}$ index collected in the $m^{th}$ vector $\mathbf{x}_m$. For simplicity, $k_{m,n}$ is assigned as

$$k_{m,n} = m + (n-1)\frac{192}{q}. \qquad (2)$$

Accordingly, the norm of vector $\mathbf{x}_m$, termed $\rho(m)$, can be derived as follows:

$$\rho(m) = |\mathbf{x}_m| = \left(\sum_{n=0}^{q-1} X(k_{m,n})X^*(k_{m,n})\right)^{1/2}, \qquad (3)$$

where the superscript $*$ denotes the complex conjugate operator. Our goal here is to determine the size of quantization step $\Delta$ which is most suitable to modulate the vector norm used in QIM. During this process, $\Delta$ ought to be adapted to the spectral intensity in order to achieve a reasonable balance between robustness and imperceptibility. Using QIM, the vector norm is modified according to the watermark bit $w(i)$, as follows:

$$\hat{\rho}(i) = \begin{cases} \left\lfloor \dfrac{\rho(i)}{\Delta} + 0.5 \right\rfloor \Delta, & \text{if } w(i) = 0; \\ \left\lfloor \dfrac{\rho(i)}{\Delta} \right\rfloor \Delta + \frac{\Delta}{2}, & \text{if } w(i) = 1. \end{cases} \qquad (4)$$

Watermarking errors, which are defined as the difference between $\hat{\rho}(i)$ and $\rho(i)$ in Eq. (4), is assumed to have a uniform distribution over $[-\Delta/2,\ \Delta/2]$. Given that the FFT coefficients and the watermarking errors maintain a power ratio $\Gamma$ in decibels, the relationship between $\Delta$ and $\Gamma$ can be estimated as follows:

$$\begin{aligned}10^{\frac{\Gamma}{10}} &= \frac{\sum_{i=1}^{L_X} X(i)X^*(i)}{L_\rho \mathrm{E}\left[\sum_{i=0}^{L_\rho-1}\left(\hat{\rho}(i)-\rho(i)\right)^2\right]}\\ &= \frac{\frac{1}{L_X}\sum_{i=1}^{L_X} X(i)X^*(i)}{\frac{1}{q}\mathrm{E}\left[\sum_{i=0}^{L_\rho-1}\left(\hat{\rho}(i)-\rho(i)\right)^2\right]}\\ &= \frac{P_X}{\frac{1}{q}\cdot\frac{\Delta^2}{12}}, \qquad (5)\end{aligned}$$

where $\mathrm{E}[\cdot]$ denotes the expectation of a random process. $P_X = \frac{1}{L_X}\sum_{i=1}^{L_X} X(i)X^*(i)$ signifies the average power of the first $L_X$ FFT coefficients. During the course of watermarking, we deliberately extend the length of the FFT coefficients from $L_X$ to $L_X + L_a$ for the purpose of maintaining a consistent power level. Hence $\Delta$ is reformulated as

$$\Delta = \left(\frac{\frac{12q}{L_X+L_a}\sum_{i=1}^{L_X+L_a} X(i)X^*(i)}{10^{\frac{\Gamma}{10}}}\right)^{1/2}. \qquad (6)$$

The connection between $\Delta$ and $\Gamma$ allows us to take psychoacoustic modeling into account while using $\Gamma$ to determine a suitable strength for watermark embedding. In accordance to the auditory masking theory [27], [28], the inserted watermark will be inaudible if the distortion energy falls below the masking threshold for each critical band. The conversion from a frequency range to a critical band in a Bark scale is carried out using the equation as below [29]:

$$z_{rep} = 13\tan^{-1}\left(0.00076 f_{rep}\right) + 3.5\tan^{-1}\left(\left(f_{rep}/7500\right)^2\right). \tag{7}$$

where $f_{rep}$ and $z_{rep}$ denote the representative frequency and resulting Bark scale, respectively. The auditory masking threshold for a band with a center Bark frequency $z_{rep}$ can be further estimated using

$$a(z_{rep}) = \lambda a_{tmn}(z_{rep}) + (1-\lambda) a_{nmn}(z_{rep}) \text{ [dB]}, \tag{8}$$

where $\lambda$ denotes the tonality factor varying between 0 and 1, $a_{tmn}(z)$ is the tone-masking noise index estimated as $a_{tmn}(z) = -0.275z - 15.025$, and $a_{nmn}(z)$ is the noise-masking noise index usually fixed as $a_{nmn}(z) = -9$ [2].

The QIM in Eq. (4) leads to variations in energy, which violates the prerequisite that $\Delta$ be retrievable from the watermarked audio signal. This conflict can be settled by minimizing the variations in energy over the $L_X$ participating coefficients and then tuning the coefficients in the index range between $L_X + 1 (= 193)$ and $L_X + L_a (= 204)$. To resolve this problem, we first sort the vector norms in descending order:

$$\rho(l_0) \geq \rho(l_1) \geq \cdots \geq \rho(l_{i-1}) \geq \rho(l_i) \geq \cdots \geq \rho(l_{L_\rho - 1}), \tag{9}$$

where $l_i$, which is drawn from $\left\{0, 1, \cdots, L_\rho - 1\right\}$, signifies the index associated the $i^{th}$ largest magnitude. When applying Eq. (4) to the $l_i^{th}$ vector norm, the optimal solution $\eta_1(l_i)$ is

$$\eta_1(l_i) = \hat{\rho}(l_i), \tag{10}$$

and the suboptimal $\eta_2(l_i)$ is

$$\eta_2(l_i) = \begin{cases} \hat{\rho}(l_i) + \Delta, & \mathit{if}\ \hat{\rho}(l_i) \leq \rho(l_i); \\ \hat{\rho}(l_i) - \Delta, & \mathit{if}\ \hat{\rho}(l_i) > \rho(l_i). \end{cases} \tag{11}$$

In general, vectors with large norms contribute more variations in energy. Conversely, vectors with small norms can be modulated directly using Eq. (4) without causing noteworthy changes in overall energy. To minimize the overall variation in energy in each frame, we select between $\eta_1(l_i)$ and $\eta_2(l_i)$ for the vector norms in the top $L_o$ ranks.

$$\left\{\hat{n}_i\right\} = \underset{\{n_i | i=0,\cdots,L_o-1\}}{\arg\min} \left| \sum_{i=0}^{L_o-1} \left(\eta_{n_i}^2(i) - \rho^2(i)\right) + \sum_{i=L_o}^{L_\rho-1} \left(\hat{\rho}^2(i) - \rho^2(i)\right) \right|, \tag{12}$$

where $L_o$ is set as 10 in this study. The two summation terms in the above equation respectively represent the energy differences in the top $L_o$ vectors and the energy differences in the remaining vectors. Output argument $\hat{n}_i$ is a binary option drawn from $\{1, 2\}$. As the final solution is drawn from $2^{L_o}$ possible combinations of $\left\{\hat{n}_i\right\}$, we pursue $\hat{n}_i$ in a brutal-force manner by choosing the best fit among $2^{L_o}$ possibilities.

Substituting $\eta_{\hat{n}_i}(i)$ for the vectors in $\{\rho(l_i) | 0 \leq i \leq L_o - 1\}$; i.e., assigning $\rho(l_i) \leftarrow \hat{\rho}_\eta(l_i) = \eta_{\hat{n}_i}(l_i)$, is meant to yield the least variation in energy. Once the modulated norm for the $l_i^{th}$ vector is determined, the FFT coefficients associated with this particular vector can be modified as follows:

$$\hat{X}(k_{l_i,n}) = X(k_{l_i,n}) \frac{\hat{\rho}_\eta(l_i)}{\rho(l_i) + \varepsilon}; \quad i = 0, 1, \cdots, L_\rho - 1; \quad n = 0, 1, \cdots, q-1, \tag{13}$$

where $\varepsilon$ represents an infinitesimal number added to the denominator to avoid having to divide by zero. To ensure a perfect match with the original energy level, we employed additional $L_a (= 12)$ FFT coefficients to absorb the energy differences which result from watermarking. The modification is formulated as

$$\hat{X}(k) = X(k) \left( \frac{\sum_{i=1}^{L_X+L_a} X(i)X^*(i) - \sum_{i=0}^{L_\rho-1} \hat{\rho}_\eta^2(i)}{\sum_{i=1}^{L_X+L_a} X(i)X^*(i) - \sum_{i=0}^{L_\rho-1} \rho^2(i)} \right)^{1/2}; \quad k = L_X + 1, \cdots, L_X + L_a. \tag{14}$$

With the use of Eq. (14), the energy in $\{\hat{X}(k) | 1 \leq k \leq L_X + L_a\}$ remains the same as that in $\{X(k) | 1 \leq k \leq L_X + L_a\}$. This renders an identical $\Delta$ while replacing $X(k)$ as $\hat{X}(k)$ in Eq. (6). Because of the symmetry of the FFT, we also need to modify $X(k)$ in the second half of the FFT sequence as follows:

$$\hat{X}(L_f - k) = \hat{X}^*(k), \quad k = 1, 2, \cdots, L_X. \tag{15}$$

The FFT coefficients in other indexes remain intact, as follows:

$$\hat{X}(k) = X(k), \quad \text{for } k = 0, L_X + L_a, \cdots, L_f - L_X - L_a - 1. \tag{16}$$

Taking the inverse FFT of $\{\hat{X}(k)\}$ renders a watermarked audio signal, termed $\hat{x}(n)$, with embedded binary information.

Embedding the watermark occasionally results in undue discontinuities at frame boundaries in the watermarked audio. The transition across frames can be smoothed using a window weighting function, as follows:

$$\hat{x}'(n) = \begin{cases} (1 - \varpi(n))\, x(n) + \varpi(n)\hat{x}(n), \\ \qquad n = 0, \cdots, 15; \\ \hat{x}(n), \quad n = 16, 17, \cdots, L_f - 17; \\ (1 - \varpi(n))\, x(n) + \varpi(n - N + 31)\hat{x}(n), \\ \qquad n = L_f - 16, \cdots, L_f - 1, \end{cases} \tag{17}$$

where $x(n)$ and $\hat{x}'(n)$ denote the original signal and the smoothed, watermarked signal, respectively. $\varpi(n)$ is

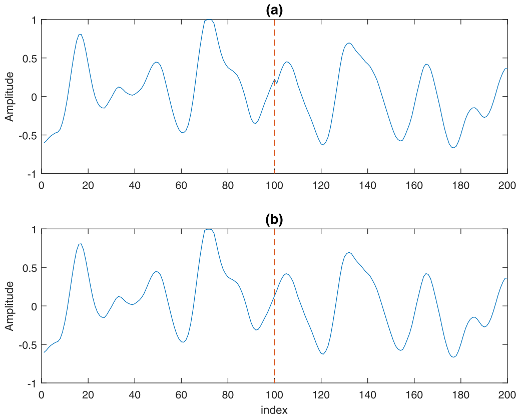


**FIGURE 2.** Effect due to window smoothing: (a) watermarked signal without the use of window smoothing at the frame boundary signified by the red dashed line. (b) smoothed watermarked signal.

a 32-point Hamming window defined as

$$\varpi(n) = 0.54 - 0.46\cos\left(2\pi\frac{n}{N_{\varpi}}\right), \quad 0 \le n \le N_{\varpi} = 31. \tag{18}$$

Figure 2 depicts the audio signals before and after window weighting. As verified in the section on performance evaluation, this slight modification does not have any detrimental effects on the watermark.

### B. AUXILIARY EMBEDDING USING IMPROVED SPREAD SPECTRUM (ISS) IN THE FFT DOMAIN

Hu *et al.* [30] recently introduced a novel SS-based audio watermarking scheme capable of providing a striking correlation peak, which facilitates watermark detection in the DCT domain. With suitable modification, the same SS-based scheme can be applied to a specified range of the FFT sequence.

The embedded target $\{\psi(k)|\, k = 0, 1, \cdots, N_{ss}\}$ is a PBS of length $N_{SS}$, with each element taking a value of "$-1$" or "1", determined at random. When inserting a PBS into the host audio, the conventional SS scheme [21] modulates the magnitude FFT as follows:

$$M_Y(k) = M_X(k)\max\{1 + \alpha\psi(k-k_b), 0\}; \quad k = k_b, k_b+1, \cdots, k_b+N_{ss}-1, \tag{19}$$

where $M_X(k) = |X(k)|$ is the magnitude of the $k^{th}$ FFT coefficient. $k_b$ denotes the beginning index, which is just the one next to the last FFT coefficient for binary embedding; i.e., $k_b = L_X + L_a + 1 = 205$. Variable $\alpha$ specifies the embedding strength, the value of which can be set to match an intended signal-to-noise ratio (SNR) in dB; i.e., $\alpha = 10^{-SNR/20}$. In accordance with the auditory masking threshold shown in Eq. (7) and (8), $SNR$ can be set at 16 [$dB$] to achieve a reasonable balance between imperceptibility and robustness [27], [28]. The modification in $M_X(k)$ takes effect on $X(k)$ via

$$Y(k) = \frac{M_Y(k)}{M_X(k)+\varepsilon}X(k), \tag{20}$$

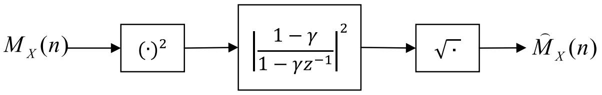


**FIGURE 3.** Procedure for smoothing the magnitude FFT coefficients.

where $Y(k)$ denotes the resultant FFT coefficient with a magnitude of $M_Y(k)$. In this study, $N_{ss}$ was set at 1024 to enable reliable detection. Given that the sampling rate is 44.1 kHz, the involved FFT coefficients occupy a frequency range from 1103.6 to 6610.7 Hz for $L_f = 8192$ and $k_b = 205$.

To detect the PBS, the linear correlation between the received FFT coefficients $\left\{\tilde{Y}(k)\right\}$ and sequence $\{\psi(k)\}$ is first computed as follows:

$$Corr\left\langle\left\{\tilde{M}_Y(k)\right\}, \{\psi(k)\}\right\rangle = \frac{1}{N_{SS}}\sum_{k=k_b}^{k_b+N_{SS}-1}\tilde{M}_Y(k)\psi(k-k_b), \tag{21}$$

where $Corr\langle\cdot, \cdot\rangle$ denotes the correlation function. The tilde symbol atop the variable implies the effect due to possible attacks. The PBS is classified as present or absent, depending on whether $Corr\langle\cdot, \cdot\rangle$ exceeds a predefined threshold. Correlation detection generally requires a relatively long sequence. For example, the time-domain SS developed in [22] requires a segment of $2^{17}$ audio samples to achieve satisfactory performance. For SS implementations in the DCT domain, the number of participating coefficients drops to approximately $2^{11}$ [23]. The length of the sequence is limited by the coefficients available in the FFT; therefore, we developed a reinforced correlation function to allow SS-based watermarking at a sequence length of 1024. As in Eq. (19), the embedding of the PBS is performed using the following:

$$M_Y(k) = \begin{cases} M_X(k) + \alpha\psi(k-k_b)\widehat{M}_X(k), & \\ \qquad \text{if } 1+\alpha\psi(k-k_b)\widehat{M}_X(k)/M_X(k) > 0.01; & \\ 0.01M_X(k), \quad \text{otherwise.} & \end{cases} \quad k = k_b, k_b+1, \cdots, k_b+N_{ss}-1; \tag{22}$$

where the second branch serves as a threshold to prevent a negative magnitude. The term $\widehat{M}_X(k)$ denotes a smoothed version of $M_X(k)$. $\widehat{M}_X(k)$ is used for the purpose of distributing the embedding strength more evenly across the frequency range, rather than concentrating on a few coefficients with large magnitudes. Thus, the resulting coefficient $M_Y(k)$ can be expressed as

$$M_Y(k) = \beta(k)M_X(k) \tag{23}$$

with

$$\beta(k) = \max\left\{0.01, \; 1+\frac{\alpha\psi(k)\widehat{M}_X(k)}{M_X(k)+\varepsilon}\right\}. \tag{24}$$

When deriving $\widehat{M}_X(k)$, we adopted a nonlinear approach shown in Fig. 3. After taking the square of each coefficient, we apply a first-order zero-phase recursive filter to produce a curve which resembles the power spectrum. The subsequent square root operation renders a smoothed version of the magnitude FFT sequence. Here, filter coefficient $\gamma$ is chosen as 0.9 to render a highly smoothed power spectrum. This makes $\widehat{M}_X(k)$ larger whenever $M_X(k)$ is close to a FFT coefficient with a large component. This arrangement coincides with the auditory masking theory, which states that noise (i.e., watermarking modification associated with $M_X(k)$) in a critical band can be masked out by a strong frequency component in the vicinity.

In view of this deficiency in conventional SS, we also incorporated a magnitude adaptation scheme within the correlation function, which results in each coefficient contributing a comparable weight to the outcome. Let magnitude FFT at the receiving end be $\tilde{M}_Y(k) = \beta(k)M_X(k) + e(k)$, i.e.

$$\begin{aligned} e(k) &= \left|\tilde{Y}(k)\right| - |Y(k)| = \tilde{M}_Y(k) - M_Y(k) \\ &= \tilde{M}_Y(k) - \beta(k)M_X(k). \end{aligned} \tag{25}$$

The linear correlation function is formulated as

$$\begin{aligned} &Corr\left\langle \left\{M'_Y(k)\right\}, \{\psi(k)\}\right\rangle \\ &= Corr\left\langle \left\{\frac{\tilde{M}_Y(k)}{\hat{M}_Y(k)}\right\}, \{\psi(k)\}\right\rangle \\ &= \frac{1}{N_{ss}}\sum_{k=k_b}^{k_b+N_{ss}-1} \frac{\beta(k)M_X(k)+e(k)}{\hat{M}_Y(k)}\psi(k) \\ &= \frac{1}{N_{ss}}\sum_{k=k_b}^{k_b+N_{ss}-1} \frac{\beta(k)M_X(k)}{\hat{M}_Y(k)}\left(1+\frac{e(k)}{\beta(k)M_X(k)}\right)\psi(k) \\ &\approx \frac{1}{N_{ss}}\sum_{k=k_b}^{k_b+N_{ss}-1} \left(1+\frac{\alpha\psi(k)\widehat{M}_X(k)}{M_X(k)+\varepsilon}\right) \\ &\quad \times \left[\left(\frac{M_X(k)}{\hat{M}_Y(k)}\right)\left(1+\frac{e(k)}{\beta(k)M_X(k)}\right)\right]\psi(k) \\ &\approx \frac{\alpha}{N_{ss}}\sum_{k=k_b}^{k_b+N_{ss}-1} \left(\frac{\widehat{M}_X(k)}{\hat{M}_Y(k)}\right)\left(1+\frac{e(k)}{\beta(k)M_X(k)}\right) \approx \alpha, \end{aligned} \tag{26}$$

where $M'_Y(k)$ is a magnitude-adjusted sequence obtained by adaptively rescaling $\tilde{M}_Y(k)$ by $1/\hat{M}_Y(k)$. Here, $\hat{M}_Y(k)$ is just another smoothed version of $\tilde{M}_Y(k)$. The value of $\hat{M}_Y(k)$ is obtained using the procedure in Fig. 3. We adopted $\gamma = 0.6$ to perform lowpass filtering, thereby rendering $\hat{M}_Y(k)$ close to $\widehat{M}_X(k)$. The use of $\gamma = 0.6$ in lowpass filtering tends to result in a moderately smoothing effect, which should not impair the synchronization information hidden in the FFT coefficients. In Eq. (26), the result within the brackets is close to unity if $|e(k)| \ll \beta(k)M_X(k)$. Hence each product term makes approximately the same contribution to the correlation function, regardless of the variation in magnitude FFT.

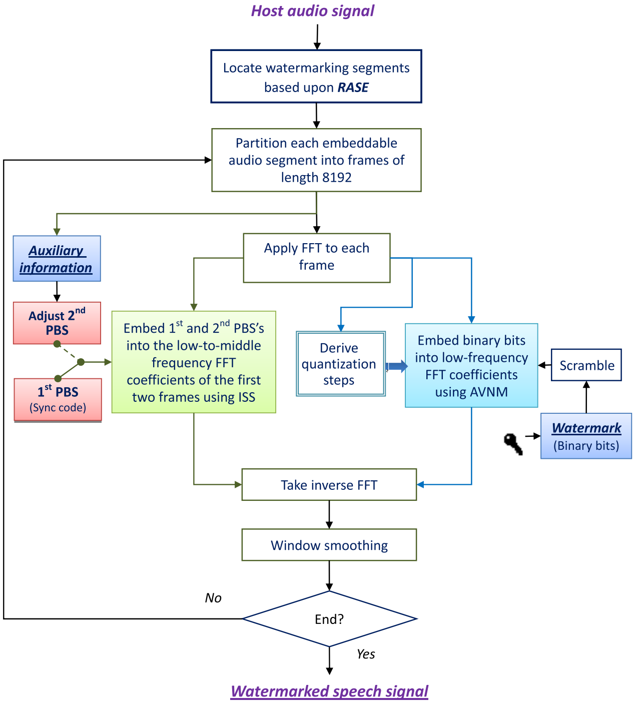

Host audio signal
Locate watermarking segments based upon RASE
Partition each embeddable audio segment into frames of length 8192
Auxiliary information
Apply FFT to each frame
Adjust 2nd PBS
Embed 1st and 2nd PBS's into the low-to-middle frequency FFT coefficients of the first two frames using ISS
Derive quantization steps
Embed binary bits into low-frequency FFT coefficients using AVNM
Scramble
1st PBS (Sync code)
Watermark (Binary bits)
Take inverse FFT
Window smoothing
No
End?
Yes
Watermarked speech signal

**FIGURE 4.** Embedding procedure of the FFT-based blind audio watermarking scheme.

## III. PROCEDURE FOR WATERMARK EMBEDDING AND EXTRACTION

Figure 4 depicts the procedure used in embedding a watermark. In the proposed scheme, the Robust Segment Extractor Algorithm (RASE) presented in [31] is used to render a series of feature points. RASE smooths the gradient audio signal using a Gaussian filter and then takes the magnitude of the smoothed gradient audio signal as an output response. The points with the largest responses are treated as preliminary feature points. RASE then goes through an elimination process involving the iterative screening out of adjacent feature points with lower responses. In this study, we set the minimum gap between any two feature points as double the frame length (i.e., $2L_f$), which means that each audio segment spans a range of no less than 16,384 samples. We then divided each audio segment into frames of $L_f$, with the leading frame starting at the RASE feature point. The frames with energy levels that exceeded the predefined threshold were selected as embeddable frames.

The role of RASE is twofold. RASE can not only locate audio frames for watermark embedding but also help to accelerate frame synchronization necessitated by the watermark extraction conducted in the FFT domain. Following frame partition, the FFT-based schemes discussed in Section II are brought in to hide various types of information in separate areas in the FFT sequence. That is,

(1) AVNM is in charge of binary embedding in low-frequency FFT coefficients $\{X(k)|\, k = 1, 2, \cdots, 204\}$. For the sake of security, the watermark is scrambled using an encryption key.

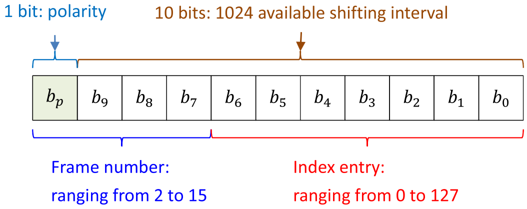
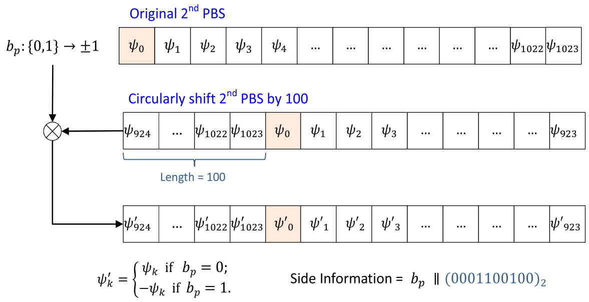


**FIGURE 5. The arrangement of the 11 bits used to carry the side information contained in an audio segment.**

(2) ISS is adopted to embed auxiliary information in low-to-middle FFT coefficients $\{X(k)| k = 205, \cdots, 1228\}$ without causing any inference with the binary embedding. The auxiliary information consists of the synchronization code for frame alignment and a circularly shifted PBS for tag indexing. Specifically, the ISS embeds the synchronization code (which is essentially a PBS) in the first frame and another circularly shifted PBS in the second frame.

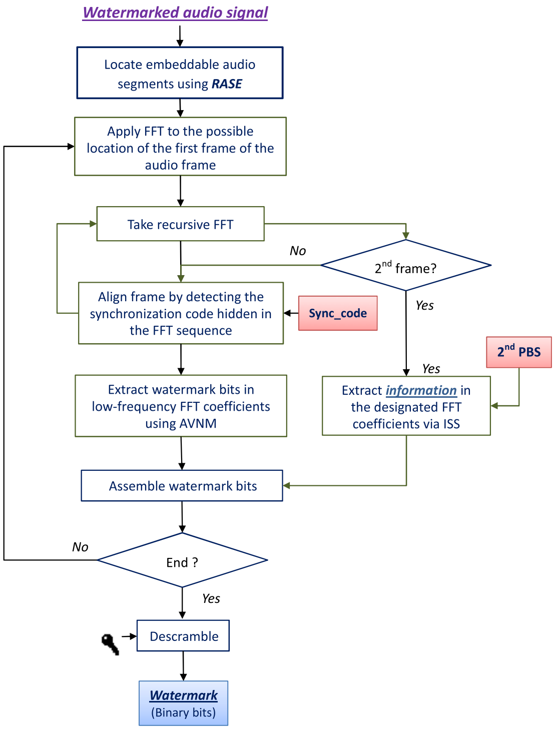
Note that other information related to the embedded PBS can also be wrapped up with SS embedding. For example, the polarity of the PBS can represent a binary variable. The shift interval of a circularly shifted PBS may indicate an integer number. Theoretically, the computational effort and distortion which results from the embedding of a circularly shifted PBS is identical with that resulting from the employment of the original PBS. In other words, the circular shift operation allows the PBS to deliver additional information at no extra cost in terms of imperceptibility or computational load in the embedding phase. However, searching for the exact location of the embedded PBS (such as the synchronization code in the first frame) can be tedious. To reduce the computation burden required to perform correlation comparisons during the search, we set the synchronization code as a fixed pattern. The correlation function is examined only once while moving the frame one sample at a time. Unlike the case of the synchronization code in the first frame, detection of a circularly shifted PSB in the second frame requires extra computation for identifying the shift interval. Fortunately, the computational burden is mild, as the frame is has already been aligned during the search for the synchronization code in the first frame. Eventually, we hid additional 11 bits (i.e., 1 bit for the polarity and 10 bits for 1024 possible shifting interval) in order to provide auxiliary information in the second frame. Figure 5 explains the formation and arrangement of the 11 bits. Among them, 4 bits are used to denote the number of available frames in an audio segment and the remaining 7 bits indicate an index entry for all the watermark bits contained in that segment. Figure 6 demonstrates how to use the PBS to deliver an 11-bit message. Given that the side information contains an 11-bit message, namely $(10001100100)_2$, we can simply circularly shift the PBS to the right by 100 and reverse the polarity of every element in the PBS.

The procedure used in watermark extraction is illustrated Fig. 7. In general, the computational power required for watermark extraction is lower than that required for embedding, due to the fact that there is no need to modify the audio signal. Nonetheless, the procedure in our case is much more complicated because we have to identify the actual position of each frame. The search for the frame boundaries imposes a heavy computational burden, particularly when the search is conducted in the transform domain. As shown in Fig. 7, RASE is first used a coarse tracker to locate the feature points in the watermarked audio. Starting from the position of the first feature point, we apply a rectangular window of length $L_f$ to extract an audio frame of the audio signal and perform FFT on that frame. We then trim the FFT sequence in order to conduct a correlation comparison, as outlined in Section II-B. This results in a value which indicates whether the synchronization code could exist there. The feature points



Original 2nd PBS
$b_p: \{0,1\} \rightarrow \pm 1$
$\psi_0$ $\psi_1$ $\psi_2$ $\psi_3$ $\psi_4$ … … … … … … $\psi_{1022}$ $\psi_{1023}$
Circularly shift 2nd PBS by 100
$\psi_{924}$ … $\psi_{1022}$ $\psi_{1023}$ $\psi_0$ $\psi_1$ $\psi_2$ $\psi_3$ … … … … $\psi_{923}$
Length = 100
$\psi'_{924}$ … $\psi'_{1022}$ $\psi'_{1023}$ $\psi'_0$ $\psi'_1$ $\psi'_2$ $\psi'_3$ … … … … $\psi'_{923}$
$\psi'_k = \begin{cases} \psi_k & \text{if } b_p = 0; \\ -\psi_k & \text{if } b_p = 1. \end{cases}$
Side Information = $b_p \parallel (0001100100)_2$

**FIGURE 6. The use of circularly shifted PBS to represent 11 bits side information. Symbol "‖" denotes the concatenation operator. $(X)_B$ denotes the numeric representation of $X$ with a base of $B$.**



Watermarked audio signal
Locate embeddable audio segments using RASE
Apply FFT to the possible location of the first frame of the audio frame
Take recursive FFT
No
2nd frame?
Yes
Align frame by detecting the synchronization code hidden in the FFT sequence
Sync_code
2nd PBS
Yes
Extract watermark bits in low-frequency FFT coefficients using AVNM
Extract information in the designated FFT coefficients via ISS
Assemble watermark bits
No
End ?
Yes
Descramble
Watermark
(Binary bits)

**FIGURE 7. Extraction procedure of the FFT-based blind audio watermarking scheme.**

selected by RASE can be disturbed by intentional attacks or unintentional modifications; therefore, extending the search range is generally appropriate. Eventually, the watermark extraction procedure involves repeatedly moving the analysis frame forward sample-by-sample, recalculating the FFT, and then comparing the magnitude FFT sequence with the synchronization code based on the correlation formula specified in Eq. (26). During each time shift, we continue updating the position of the largest absolute value in the correlation function. The search in the forward direction terminates when the distance moved is 16 samples from the largest absolute value. The same search process is then applied backward to ensure that the location of the largest absolute value corresponds to a peak. The foregoing search process starts with the acquisition of the FFT at a pace of one sample per iteration, thereby imposing a heavy computational load on watermark extraction. Fortunately, the FFT sequence of a frame can be computed recursively from a frame that is one sample ahead or behind.

Let $X_p(k)$ denote the FFT coefficient obtained from an audio frame of length $N(= L_f)$ starting at position $p$.

$$X_p(k) = \sum_{n=0}^{N-1} x(p+n)W^{nk} \tag{27}$$

with

$$W = e^{-j\frac{2\pi}{N}}. \tag{28}$$

Thanks to the recursive formulation of the FFT, we can derive a new FFT sequence from the available ones when the analysis frame is moved forward (i.e., from $p$ to $p+1$) by one sample.

$$\begin{aligned} X_{p+1}(k) &\\ = \sum_{n=0}^{N-1} x(p+n+1)W^{nk} &\underset{(m=n+1)}{=} \sum_{m=1}^{N} x(p+m)W^{(m-1)k} \\ = \left[-x(p)W^0 + x(p+N)W^{Nk} + \sum_{m=0}^{N-1} x(p+m)W^{mk}\right] W^{-k} &\\ = \left[-x(p) + x(p+N) + X_p(k)\right] W^{-k}. & \end{aligned} \tag{29}$$

The operation involves replacing $x(p)$ with $x(p+N)$ in the formulation, followed by the multiplication of $W^{-k}$ for the $k^{th}$ coefficient. Similarly, when the analysis frame moves backward, the new FFT sequence becomes

$$\begin{aligned} X_{p-1}(k) &\\ = \sum_{n=0}^{N-1} x(p+n-1)W^{nk} &\underset{(m=n-1)}{=} \sum_{m=-1}^{N-2} x(p+m)W^{(m+1)k} \\ = x(p-1)W^0 - x(p+N-1)W^{Nk} + \sum_{m=0}^{N-1} x(p+m)W^{mk} \cdot W^k &\\ = x(p-1) - x(p+N-1) + X_p(k)W^k. & \end{aligned} \tag{30}$$

Using Eq. (30), the steps described in the forward process are applied to the watermarked signal in the backward direction one sample at a time. The above processes repeat until the search finds a salient local peak. In the event that the peak value of the correlation function exceeds a predefined threshold, then its position is set to the beginning of the frame. If the thresholding criterion is not met, then the synchronization code is assumed to be absent. The extraction routine then jumps directly to the next feature point and launches a new search. Figure 8 presents a typical example of frame partition and synchronization. Following frame partition and synchronization, we extracted the index tag hidden in the second frame. Figure 9 exemplifies the information recovered from the second frame of each audio segment. The derived information instructs the FFT-AVNM scheme to retrieve binary bits from the embedded frames. Let $\tilde{X}_{\hat{p}}(k)$ denotes the $k^{th}$ FFT coefficient obtained from a watermarked frame at position $\hat{p}$. Substituting $\tilde{X}_{\hat{p}}(k)$ for $X(k)$ in Eqs. (3) and (6) results in the vector norm $\tilde{\rho}(i)$ and adaptive quantization step $\tilde{\Delta}$, respectively. The $i^{th}$ watermark bit, termed $\tilde{w}(i)$, is then determined based on the QIM rule:

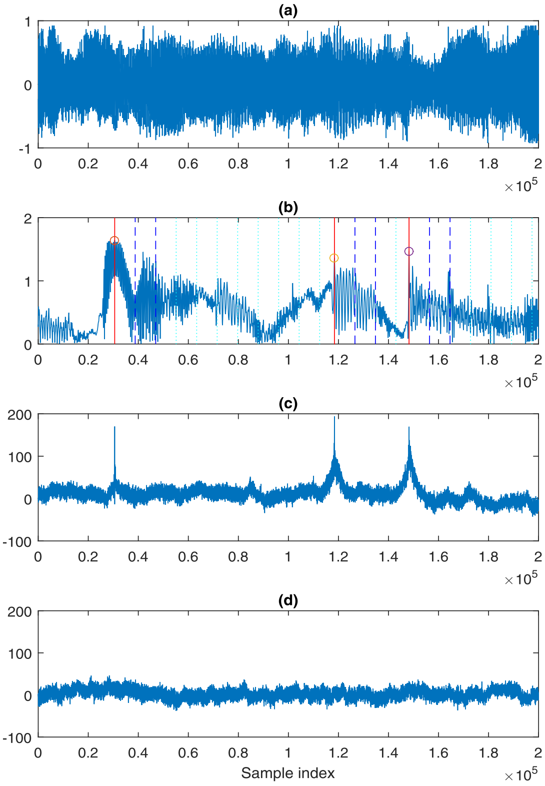


**FIGURE 8.** Illustration of synchronization code detection: (a) Time waveform of the audio signal, (b) Response of the RASE; The "o"-shape markers represent the feature points, each of which can be used to indicate the beginning of an embeddable audio segment. (c) Enhanced correlation function using a correct PBS in the FFT domain when the analysis window moves across time, (d) the result using an incorrect PBS.

$$\tilde{w}(i) = \begin{cases} 1, & \text{if } \left| \dfrac{\tilde{\rho}(i)}{\tilde{\Delta}} - \left\lfloor \dfrac{\tilde{\rho}(i)}{\tilde{\Delta}} \right\rfloor - 0.5 \right| \le 0.25; \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

The next step is to piece all the retrieved watermark bits together as a whole. If multiple copies of the bits are available,

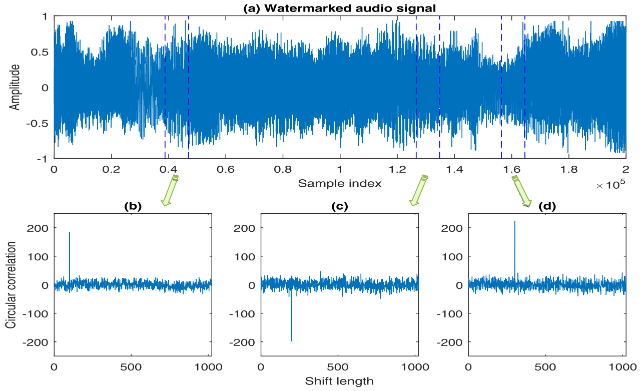


**FIGURE 9.** **Illustration of information retrieval from the second frame (demarcated by two blue dashed line) within each embeddable audio segment. The index with the sharpest peak (either positive or negative) represents the embedded integer number, which is 100, 200 and 300 in (b), (c), and (d), respectively.**

then majority voting is used to make the final decision. The above process continues until all feature points have been used. Finally, once all of the binary bits have been gathered together, the watermark is descrambled using the encryption key.

## IV. PERFORMANCE EVALUATION

The test materials in the following experiments comprised sixty 30-second music clips drawn from RWC (Real World Computing) Music Genre Database [32]. The music clips could be classified within the following nine categories: popular (5), rock (5), dance (5), jazz (4), Latin (6), classical (14), march (4), world (13), and vocal (4). All audio signals were sampled at 44.1 kHz with 16-bit resolution. While testing various schemes with different payload capacities, the watermark bits used in the test were a series of alternating 1's and 0's long enough to cover the entire host signal. The synchronization code was a PBS of length 1024 with an equal number of "+1" and "−1" values. We used a second PBS to record an 11-bit message shown in Fig. 5, which includes 7 bits for the index entry of the watermark bits and 4 bits for the frame count in a given audio segment. The message was inserted by circularly shifting the polarized PBS to a specific location corresponding to a 10-bit number.

**TABLE 1.** **Processing times (in seconds) required for carrying out watermark embedding and extraction.**

| CPU time | Embedding procedure | | | Extraction procedure | | |
|---|---|---|---|---|---|---|
| | RASE | Sync_code & PBS embedding | Binary embedding | RASE | Sync_code & PBS detection | Binary extraction |
| Mean | 0.324 | 0.075 | 1.761 | 0.310 | 4.394 | 0.047 |
| Standard deviation | 0.136 | 0.016 | 0.069 | 0.101 | 3.915 | 0.002 |

### A. PROCESSING COMPLEXITY

In this study, we implemented the proposed blind audio watermarking in a Matlab environment operating with an Intel I7-4790 CPU and 32G RAM. Table 1 presents the average processing times required for executing an audio file of 30-second long. Most of the CPU time was expended on the detection of embedded synchronization codes. The required CPU time varied with the examined audio and the attack inflicted on that audio. The reason is due to that the attack may perturb the accuracy of the RASE, which affects the initial estimate of the locations and quantities of the synchronization codes in an audio. This also explains why the required time in the synchronization code detection has a relatively large standard deviation.

### B. DETECTION OF DESIGNATED PBS PATTERN IN FFT SEQUENCES

Our second concern was the survivability of the synchronization codes inserted in the FFT sequence between the 205th and 1228th coefficients using the formula in Eq. (22). In addition to the implementation of ISS, we also adopted conventional SS [21] (denoted as CSS) for evaluation and comparison purposes.

The quality of the resulting watermarked audio signals was assessed using signal-to-noise ratio (SNR), as defined in Eq. (32), in conjunction with the PEAQ metric [33].

$$SNR = 10\log_{10}\left(\frac{\sum_{n} x^2(n)}{\sum_{n}\left(\hat{x}'(n)-x(n)\right)^2}\right), \tag{32}$$

where $x(n)$ and $\hat{x}'(n)$ follow the same definitions in Eq. (17). The PEAQ metric was an implementation released by the TSP Lab at McGill University [33]. It renders an objective difference grade (ODG) between −4 and 0, signifying a perceptual impression from "very annoying" to "imperceptible". Table 2 lists the five-grade quality scale of PEAQ.

**TABLE 2.** **Five-grade quality scale of PEAQ.**

| Impairment description | ODG |
|---|---|
| Imperceptible | 0.0 |
| Perceptible, but not annoying | −1.0 |
| Slightly annoying | −2.0 |
| Annoying | −3.0 |
| Very annoying | −4.0 |

**TABLE 3.** **Statistics of the measured SNR's and ODG's. The data in the last two columns are interpreted as "mean [standard deviation]".**

| Scheme | | FFT-CSS | FFT-ISS |
|---|---|---|---|
| SNR | Embedded segments only | 26.56 [3.29] | 26.61 [3.20] |
| | Entire audio signal | 29.90 [2.92] | 29.98 [2.84] |
| ODG | Embedded segments only | -0.17 [0.11] | -0.17 [0.12] |
| | Entire audio signal | -0.04 [0.14] | -0.05 [0.15] |

Table 3 presents SNRs and ODGs measured in this study. These two measures were used to assess the watermarked segments as well as the entire audio clip. As revealed in the tabulated data, the average SNR values obtained using the two SS-based schemes far exceeded the preset SNR value (i.e., 16 dB). This is due to fact that the preset SNR is only applied to designated FFT coefficients, whereas SNR takes account of all FFT coefficients. The average ODGs in the embedded audio segments were above −0.2 (for both schemes), which implies that the original and watermarked audio signals were nearly indistinguishable. When the measurement duration was extended to cover the entire audio clip, the overall average SNR and ODG increased to 29.98 dB and −0.05, respectively. The near zero ODG values suggest that it would be highly unlikely that any listener would notice a difference between the watermarked audio and the original.

Interpreting the message in the second PBS required multiple correlation comparisons; therefore, we resorted to FFT to compute the circular correlation between $\left\{\tilde{M}_y(k)/\hat{M}_y(k)\right\}$ and $\{\psi(k)\}$. More specifically, we first applied FFT to the two sequences individually, and then multiplied one sequence by the complex conjugate of the other in an element-wise manner. Taking the inverse FFT of the multiplication result renders a circular correlation function $R(n)$ with $n$ denoting the circular shift.

$$\begin{aligned}&\{R(n)|\, n = 0, 1, \cdots, N_{ss}-1\}\\&\quad = \mathcal{F}^{-1}\left\{\frac{1}{N_{ss}}\mathcal{F}\left\{\tilde{M}_y(k)/\hat{M}_y(k)\right\} \circ (\mathcal{F}\{\psi(k)\})^*\right\}, \qquad (33)\end{aligned}$$

where $\mathcal{F}\{\cdot\}$ and $\mathcal{F}^{-1}\{\cdot\}$ denote the FFT and inverse FFT operations, respectively. Symbol "$\circ$" represents the element-wise multiplication. The zero lag of $R(n)$ (i.e., $R(0)$) coincides with the correlation function associated with the synchronization code, and the results of other lags emulate the random test of the linear correlation using arbitrary PBS.

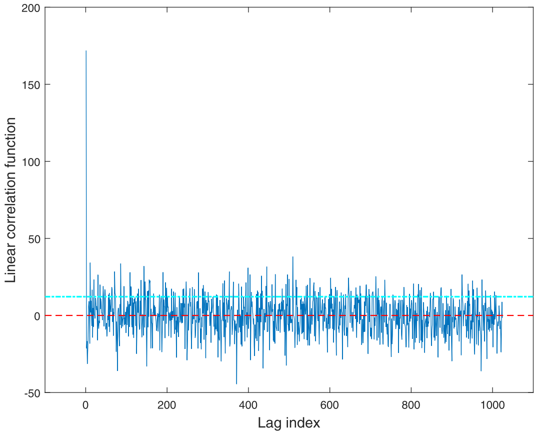


**FIGURE 10.** A typical example of the linear correlation function obtained by the proposed scheme. The cyan dash-dot line stands for the RMS of the correlation function drawn from the indexes other than zero.

Figure 10 delineates a typical linear correlation function. In accordance with the SS formulation, a large value is expected to occur at index zero. The values appearing in other indexes exhibit noise-like fluctuations around zero. The synchronization code can be identified as long as the value at index zero is (1) the largest of all and (2) preferably above a predefined threshold. To evaluate the performance of the embedding schemes, we defined a measure $\rho_c$, which is the ratio of the correlation function at index zero to the root-mean-square values associated with other indexes (referred to as "contrast ratio"):

$$\rho_c = \frac{R(0)}{\sqrt{\sum_{n=1}^{N_{ss}-1} R^2(n)}}. \qquad (34)$$

Theoretically, a higher contrast ratio indicates that fewer false detections can happen.

The detectability of the embedded watermarks was evaluated by comparing the resulting contrast ratios in the presence of frequently encountered attacks. These attacks included resampling, requantization, amplitude scaling, noise corruption, lowpass filtering with different cutoff frequencies, DA/AD conversion, echo addition, MPEG-I layer 3 compression, and time shifting. Table 4 outlines the details of these attacks.

**TABLE 4.** Attack types and specifications.

| Item | Type | Description |
|---|---|---|
| A | Resampling | Conduct down-sampling to 22050 Hz and then up-sampling back to 44100 Hz. |
| B | Requantization | Quantize the watermarked signal to 8 bits/sample and then back to 16 bits/sample. |
| C | Amplitude scaling | Scale the amplitude of the watermarked signal by 0.85. |
| D | Zero thresholding | Zero out all samples below a threshold, which is set as 0.03 of the maximum dynamic range. |
| E | Noise corruption (I) | Add zero-mean white Gaussian noise to the watermarked audio signal with SNR = 30 dB. |
| F | Noise corruption (II) | Add zero-mean white Gaussian noise to the watermarked audio signal with SNR = 20 dB. |
| G | Lowpass filtering (I) | Applying a lowpass filter with a cutoff frequency of 8 kHz. |
| H | Lowpass filtering (II) | Applying a lowpass filter with a cutoff frequency of 4 kHz. |
| I | DA/AD conversion | Convert the digital audio file to an analog signal and then resampling the analog signal at 44.1 kHz. The DA/AD conversion is performed through an onboard Realtek ALC892 audio codec, of which the line-out is linked with the line-in using a cable line during playback and recording. |
| J | Echo addition | Add an echo signal with a delay of 50 ms and a decay to 5% to the watermarked audio signal. |
| K | Jittering | Deleting or adding one sample randomly for every 100 samples within each frame. |
| L | MPEG-1 layer 3 compression @ 128 kbps | Compress and decompressing the watermarked audio signal with a MPEG-1 layer 3 coder at a bit rate of 128 kbps. |
| M | MPEG-1 layer 3 compression @ 64 kbps | Compress and decompressing the watermarked audio signal with a MPEG-1 layer 3 coder at a bit rate of 64 kbps. |
| N | Time shift by 1 sample | Purposely shifting the watermarked audio signal by one sample. |
| O | Time shift by 3 samples | Purposely shifting the watermarked audio signal by three samples. |

As shown in Table 5, the proposed ISS yielded values which were significantly higher than those obtained using CSS in all cases considered in this study. The data combination in each cell of Table 5 can also be interpreted as the mean $\mu_s$ and standard deviation $\sigma_s$ of the sampling distribution when the event is present. Suppose that $\{R(n)|\, n = 1, 2, \cdots, 1023\}$ presents a Gaussian distribution with a zero mean (i.e., $\mu_n = 0$) and a unit variance ($\sigma_n = 1$). We use $\rho_c > \lambda$ as the criterion by which to determine the presence of the PBS. The probability of failing to detect the

**TABLE 5.** Statistics of the measured contrast ratios, $\rho_c'$s. The expression in each data cell follows the definition in Table 3.

| Attack type | FFT-CSS | FFT-ISS |
|---|---|---|
| 0 | 4.14 [1.24] | 13.52 [1.59] |
| A | 4.14 [1.24] | 13.52 [1.59] |
| B | 4.13 [1.25] | 12.84 [1.24] |
| C | 4.14 [1.24] | 13.52 [1.59] |
| D | 4.14 [1.24] | 13.52 [1.59] |
| E | 4.10 [1.25] | 12.05 [1.25] |
| F | 3.93 [1.25] | 9.40 [1.85] |
| G | 4.14 [1.24] | 13.52 [1.59] |
| H | 3.21 [0.75] | 10.02 [1.27] |
| I | 4.03 [1.26] | 10.83 [1.54] |
| J | 4.14 [1.24] | 13.49 [1.58] |
| K | 3.90 [1.16] | 9.67 [0.60] |
| L | 4.11 [1.22] | 13.16 [1.55] |
| M | 3.87 [1.10] | 10.43 [1.26] |
| N | 4.13 [1.24] | 13.23 [1.37] |
| O | 4.08 [1.27] | 11.94 [1.47] |

**TABLE 6.** Miss and false alarm rates estimated from the statistics presented in Table 5.

| Attack type | FFT-CSS | | FFT-ISS | |
|---|---|---|---|---|
| | $P_{Miss}$ | $P_{FA}$ | $P_{Miss}$ | $P_{FA}$ |
| 0. none | 1.51E-02 | 7.38E-02 | 1.67E-08 | 1.11E-06 |
| A | 1.51E-02 | 7.38E-02 | 1.66E-08 | 1.11E-06 |
| B | 1.57E-02 | 7.44E-02 | 9.62E-12 | 3.51E-06 |
| C | 1.51E-02 | 7.38E-02 | 1.67E-08 | 1.11E-06 |
| D | 1.51E-02 | 7.38E-02 | 1.67E-08 | 1.11E-06 |
| E | 1.66E-02 | 7.58E-02 | 1.63E-10 | 1.24E-05 |
| F | 2.08E-02 | 8.47E-02 | 4.95E-04 | 5.03E-04 |
| G | 1.51E-02 | 7.38E-02 | 1.66E-08 | 1.11E-06 |
| H | 2.62E-03 | 1.30E-01 | 1.42E-07 | 2.26E-04 |
| I | 1.86E-02 | 7.91E-02 | 2.41E-06 | 7.55E-05 |
| J | 1.51E-02 | 7.38E-02 | 1.36E-08 | 1.17E-06 |
| K | 1.42E-02 | 8.60E-02 | 3.73E-26 | 3.59E-04 |
| L | 1.45E-02 | 7.50E-02 | 1.82E-08 | 2.04E-06 |
| M | 1.11E-02 | 8.81E-02 | 3.55E-08 | 1.31E-04 |
| N | 1.55E-02 | 7.42E-02 | 1.52E-10 | 1.81E-06 |
| O | 1.84E-02 | 7.68E-02 | 7.19E-08 | 1.47E-05 |

PBS, termed $P_{Miss}$, can be estimated as follows:

$$P_{Miss} = \Phi\left(\frac{\mu_s - \lambda}{\sigma_s}\right) \tag{35}$$

with function $\Phi(\cdot)$ defined as $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{x^2}{2}} dx$. The false alarm rate $P_{FA}$ is the probability of falsely reporting the presence of the PBS. It can be calculated as follows:

$$P_{FA} = 1 - \Phi\left(\frac{\lambda - \mu_n}{\sigma_n}\right) = 1 - \Phi(\lambda). \tag{36}$$

In the case that $\lambda$ is assigned as $0.35\rho_c$, Table 6 provides the estimated $P_{Miss}$and $P_{FA}$ values for the parameters drawn from Table 5. Table 6 shows that the improved SS resulted in far lower miss rates than did CSS under all types of attacks. By contrast, CSS suffered much worse miss rates around 0.02 and false alarm rates around 0.08 under these attacks.

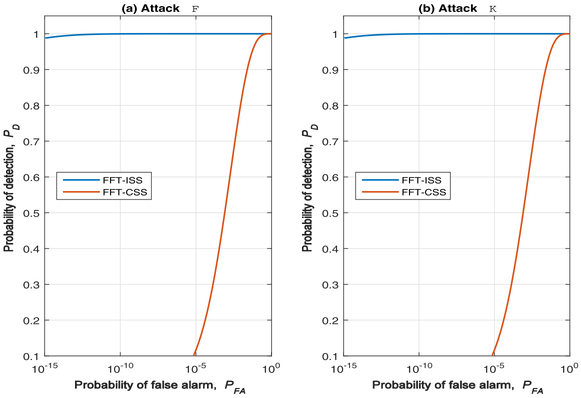


**FIGURE 11.** The receiver operating characteristic (ROC) curves corresponding to attacks $F$ and $K$.

**TABLE 7.** Statistics of the measured SNR's and ODG's along with the payload capacities. The data expressions in the second and third columns follow the definition in Table 3.

| Watermarking schemes | SNR | ODG | Payload (bps) |
|---|---|---|---|
| DWT–KFDA | 28.63 [2.69] | -1.19 [1.28] | 344.53 |
| DWT–RDM | 20.57 [1.10] | -0.21 [0.31] | 344.53 |
| FFT–LR | 16.04 [1.28] | -0.57 [0.48] | 344.53 |
| DWT–DCT | 20.91 [1.20] | -0.25 [0.65] | 602.93 |
| DWT–VDVM | 20.89 [0.51] | -0.26 [0.36] | 602.93 |
| DCT | 17.57 [0.69] | -0.25 [0.36] | 848.08 |
| FFT–AVNM-(I) | 19.93 [0.70] | -0.20 [0.29] | 344.53 |
| FFT–AVNM-(II) | 20.31 [0.67] | -0.23 [0.30] | 689.06 |
| FFT–AVNM-(III) | 20.77 [0.66] | -0.39 [0.48] | 1033.59 |

Figure 11 depicts the receiver operating characteristic (ROC) curves corresponding to the cases considered in attacks $F$ and $K$. These two cases are chosen for demonstration because they have lowest gaps between the two mean values respectively deduced from FFT-CSS and FFT-ISS. In both cases, the ROC curves for the FFT-ISS scheme are much closer to the upper left corner than do those for the FFT-CSS scheme, indicating that FFT-ISS can provide much better detection accuracy.

### C. COMPARATIVE EVALUATION OF IMPERCEPTIBILITY AND ROBUSTNESS

The parameters used in FFT-AVNM were set as follows: $L_f = 8192$, $L_X = 204$, $L_a = 12$, $\Gamma = 21$. This study considered three versions of FFT-AVNM with different payload capacities: (1) 344.53 bits per second (bps) for FFT-AVNM-(I) with $q$ set at 3, (2) 689.06 bps for FFT-AVNM-(II) with $q$ set at 2, and (3) 1033.59 bps for FFT-AVNM-(III) with $q$ set at 1. We compared the performance of the proposed FFT-AVNM scheme in terms of imperceptibility and robustness against six other schemes, namely DWT-KFDA [14], DWT-RDM [13], FFT-LR [8], DWT-DCT [5], DWT-VDVM [15], and DCT [4] in abbreviated form. These six schemes were selected because their capacities are within the test range of FFT-AVNMs. In FFT-LR, the frame size was set to 8192 to match that of FFT-AVNM. The frequency samples used for

**TABLE 8.** Averaged bit error rates (in percentage) of the extracted watermarks obtained from the compared schemes.

| Attack type | DWT–KFDA | DWT–RDM | FFT–LR | DWT–DCT | DWT–VDVM | DCT | FFT–AVNM-(I) | FFT–AVNM-(II) | FFT–AVNM-(III) |
|---|---|---|---|---|---|---|---|---|---|
| 0. None | 0.00 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 | **0.00** | 0.00 | 0.00 |
| A | 0.00 | 0.00 | 0.00 | 0.11 | 0.00 | 0.00 | **0.00** | 0.00 | 0.00 |
| B | 0.00 | 0.00 | 0.52 | 0.12 | 0.01 | 0.04 | **0.00** | 0.00 | 0.00 |
| C | 0.00 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 | **0.00** | 0.00 | 0.00 |
| D | 0.00 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 | **0.00** | 0.00 | 0.00 |
| E | 0.00 | 0.00 | 1.67 | 0.16 | 0.03 | 0.18 | **0.00** | 0.00 | 0.00 |
| F | 3.11 | 0.07 | 9.52 | 1.02 | 0.45 | 1.76 | **0.06** | 0.13 | 0.46 |
| G | 0.00 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 | **0.00** | 0.00 | 0.00 |
| H | 41.56 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 | **0.00** | 0.00 | 0.00 |
| I | 0.33 | **0.03** | 4.60 | 0.67 | 0.12 | 0.62 | 0.20 | 0.25 | 0.35 |
| J | 0.07 | 0.18 | **0.01** | 4.31 | 1.34 | 3.32 | 2.71 | 3.27 | 3.38 |
| K | 9.38 | 0.22 | 11.67 | 0.47 | 0.85 | 2.24 | **0.13** | 0.25 | 0.84 |
| L | 0.19 | 0.01 | 2.22 | 0.13 | 0.02 | 0.05 | **0.00** | 0.00 | 0.03 |
| M | 4.13 | 1.33 | 17.03 | 2.64 | 3.44 | 4.43 | **1.20** | 2.40 | 5.61 |
| N | 17.24 | 0.42 | 0.00 | 0.81 | 1.49 | 2.57 | **0.00** | 0.00 | 0.00 |
| O | 72.61 | 7.61 | 0.00 | 7.33 | 20.96 | 16.60 | **0.00** | 0.00 | 0.00 |

watermarking in FFT-LR were also consistent with those used in FFT-AVNM. The linear regression used for each binary FFT-LR watermark was applied to three adjacent FFT coefficients, which led to a capacity of 344.53 bps. As for the other schemes in the comparison, the parameters were the same as those specified in the literature.

We evaluated the quality of the watermarked audio signal based on the *SNR* defined using Eq. (32) and PEAQ [33]. As shown in Table 7, DWT-KFDA achieved the highest SNR; however, this scheme showed the worst average ODG value. This can be explained by the fact that DWT-KFDA does not apply congruous embedding strength across the DWT coefficients. DCT achieved the second lowest SNR, and the resulting ODG was among the lowest in the group. The excellent performance of the DCT scheme can be attributed to the fact that watermark embedding was subject to an auditory masking constraint. Among all of the schemes considered in the comparison, FFT-LR presented the lowest SNR; however, the resulting average ODG was not as unsatisfactory as expected. This can be partly attributed to the fact that FFT-LR attempts to alter adjacent FFT components based on linear regression. The ODGs achieved by the DWT-DCT, DWT-VDVM, and DWT-RDM schemes were very close to zero, indicating that the watermarked audio signals were perceptually indistinguishable from the original signals. The SNRs of the three FFT-AVNMs were around 20 dB, which is slightly less than the specification (i.e., $\Gamma = 21$). Such an outcome is due to the contribution of the suboptimal QIM used in Eq. (12). Careful inspection revealed that FFT-AVNM-(I), a scheme which features a lower bitrate, tends to have a lower SNR as well. This is conceivably due to the use of an equal number of vectors (10 in this study) to maintain energy balance. Assuming that a vector has an equal chance of changing from an optimal to suboptimal QIM, FFT-AVNM-(I) is affected the most in terms of the total number of FFT coefficients, followed by FFT-AVNM-(II) and FFT-AVNM-(III). Surprisingly, the slight decrease in SNR did not appear to compromise imperceptibility. Though the difference in ODG is not manifest, the average ODG actually increased from $-0.39$ for FFT-AVNM-(III) to $-0.20$ for FFT-AVNM-(I).

We evaluated robustness against attacks by examining the bit error rate (BER) between the recovered watermark $\tilde{W} = \{\tilde{w}(n)\}$ and the original watermark $W = \{w(n)\}$:

$$BER\left(W, \tilde{W}\right) = \frac{\sum_{n=1}^{N_w} w(n) \oplus \tilde{w}(n)}{N_w}, \tag{37}$$

where $N_w$ denotes the total number of bits. The attacks considered in this evaluation were those listed in Table 4.

Table 8 lists the average BERs obtained using the watermarking schemes investigated in this study. All of the schemes except for DWT-DCT succeeded in restoring the watermarks in the absence of attacks. Note that DWT-DCT performed QIM using a quantization step acquired directly from the DWT coefficients. The problem with the DWT-DCT scheme is that it lacks a mechanism to compensate for the excessive alteration caused by watermarking. Except DWT-DCT, all of the watermarking schemes survived requantization, resampling, amplitude scaling, and zero thresholding attacks. In the case of lowpass filtering with a cutoff frequency of 8 kHz, all schemes performed perfectly. DWT-KFDA was the only one failed when the cutoff frequency was decreased to 4 kHz. This can be attributed to the fact that the watermark was embedded in a subband spanning 0 to 5.5125 kHz. Therefore, setting the cutoff frequency to 4 kHz caused an unrecoverable loss to DWT-KFDA.

All of the schemes except FFT-LR proved highly effective in cases of noise corruption where SNR = 30 dB; however, the performance DWT-KFDA degraded when the SNR dropped to 20 dB. DA/AD conversion can be regarded as a composite effect of time-scaling, amplitude scaling, and noise corruption [34]; therefore, FFT-LR did not perform well in DA/AD conversion. The minor time shift appeared not to cause any harm to the FFT-LR or FFT-AVNM schemes, due to the fact that FFT computation employed a far wider range of audio samples than did the other schemes. Jittering attacks inflicted serious damage to DWT-KFDA and FFT-LR; however, it had little effect on the DWT-RDM, DWT-DCT, DWT-VDVM, and DCT schemes as well as on the proposed FFT-AVNM scheme. Schemes with high capacities (such as those considered in the performance comparison) are susceptible to the effects of MPEG compression. In the case of 128 kbps MPEG-1 layer 3 compression, FFT-AVNM-(I) and FFT-AVNM-(II) successfully retrieved watermarks without any errors. They also presented good robustness against the 64 kbps MPEG-1 layer 3 codec.

Overall, FFT-AVNM demonstrated excellent resistance against most of the attacks; however, performance against echo addition and 64 kbps MPEG-1 layer 3 compression was only passable. Nonetheless, even in the worse scenario (i.e., FFT-AVNM-(III) under the attack by 64 kbps MPEG-1 layer 3 compression), the average BER of 5.61% was sufficient to verify the credibility of the watermark. Finally, the data in the last three columns of Table 8 illustrate a principle commonly observed in blind audio watermarking. Specifically, for a given embedding strength (in terms of SNR) there is a trade-off between robustness (in terms of BER) and payload capacity (in terms of bps).

## V. CONCLUSIONS

This study presents a novel FFT-based blind audio watermarking method, which employs two schemes to embed two types of binary information within separate sections of an FFT sequence. Binary embedding is achieved by applying a scheme called ''adaptive vector norm modulation (AVNM)'' to low frequency FFT coefficients, while auxiliary data such as the synchronization code and index tag are inserted in low-to-middle FFT coefficients using a spread spectrum (SS)-based approach. The improved SS technique makes it possible to accurately identify the synchronization code within a range of 1024 FFT coefficients and thereby retrieve embedded auxiliary information. The incorporation of RASE to search for embedding locations facilitates self-synchronous watermarking. The study demonstrates three versions of FFT-AVNM which respectively operate at 344.53, 689.06, and 1033.59 bps. Experiment results indicate that the proposed FFT-AVNM is capable of attaining an ODG score around −0.3 with an SNR of approximately 20 dB. Furthermore, FFT-AVNM proved highly robust against a wide range of common signal processing attacks. The proposed scheme was able to match or outperform six well-established blind audio watermarking schemes when evaluated according to a composite measure that included robustness, imperceptivity, and payload capacity.

## REFERENCES

[1] N. Cvejic and T. Seppanen, *Digital Audio Watermarking Techniques and Technologies: Applications and Benchmarks*. Hershey, PA, USA: Information Science Reference, 2008.

[2] X. He, *Watermarking in Audio: Key Techniques and Technologies*. Youngstown, NY, USA: Cambria Press, 2008.

[3] G. Hua, J. Huang, Y. Q. Shi, J. Goh, and V. L. L. Thing, "Twenty years of digital audio watermarking—A comprehensive review," *Signal Process.*, vol. 128, pp. 222–242, Nov. 2016.

[4] H.-T. Hu and L.-Y. Hsu, "Robust, transparent and high-capacity audio watermarking in DCT domain," *Signal Process.*, vol. 109, pp. 226–235, Apr. 2015.

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

[6] Y. Xiang, I. Natgunanathan, S. Guo, W. Zhou, and S. Nahavandi, ''Patchwork-based audio watermarking method robust to de-synchronization attacks,'' *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 9, pp. 1413–1423, Sep. 2014.

[7] H.-H. Tsai, J.-S. Cheng, and P.-T. Yu, "Audio watermarking based on HAS and neural networks in DCT domain," *EURASIP J. Adv. Signal Process.*, vol. 2003, no. 3, pp. 252–263, 2003. doi: 10.1155/S1110865703208027.

[8] M. Fallahpour and D. Megias, "High capacity robust audio watermarking scheme based on FFT and linear regression," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 4, pp. 2477–2489, 2012.

[9] D. Megías, J. Serra-Ruiz, and M. Fallahpour, "Efficient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification," *Signal Process.*, vol. 90, no. 12, pp. 3078–3092, 2010.

[10] R. Tachibana, S. Shimizu, S. Kobayashi, and T. Nakamura, "An audio watermarking method using a two-dimensional pseudo-random array," *Signal Process*, vol. 82, no. 10, pp. 1455–1469, 2002.

[11] W. Li, X. Xue, and P. Lu, "Localized audio watermarking technique robust against time-scale modification," *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 60–69, Feb. 2006.

[12] X. Wang, P. Wang, P. Zhang, S. Xu, and H. Yang, "A norm-space, adaptive, and blind audio watermarking algorithm by discrete wavelet transform," *Signal Process.*, vol. 93, no. 4, pp. 913–922, 2013.

[13] H.-T. Hu and L.-Y. Hsu, "A DWT-based rational dither modulation scheme for effective blind audio watermarking," *Circuits, Syst., Signal Process.*, vol. 35, no. 2, pp. 553–572, 2016.

[14] H. Peng, B. Li, X. Luo, J. Wang, and Z. Zhang, "A learning-based audio watermarking scheme using kernel Fisher discriminant analysis," *Digit. Signal Process.*, vol. 23, no. 1, pp. 382–389, 2013.

[15] 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," *Digit. Signal Process.*, vol. 31, pp. 115–123, Aug. 2014.

[16] V. Bhat, I. Sengupta, and A. Das, "An adaptive audio watermarking based on the singular value decomposition in the wavelet domain," *Digit. Signal Process.*, vol. 20, no. 6, pp. 1547–1558, 2010.

[17] B. Y. Lei, I. Y. Soon, and Z. Li, "Blind and robust audio watermarking scheme based on SVD–DCT," *Signal Process.*, vol. 91, no. 8, pp. 1973–1984, 2011.

[18] H.-T. Hu *et al.*, "Incorporation of perceptually adaptive QIM with singular value decomposition for blind audio watermarking," *EURASIP J. Adv. Signal Process.*, vol. 2014, no. 12, pp. 1–12, 2014. doi: 10.1186/1687-6180-2014-12.

[19] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[20] P. Moulin and R. Koetter, "Data-hiding codes," *Proc. IEEE*, vol. 93, no. 12, pp. 2083–2126, Dec. 2005.

[21] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[22] P. Bassia, I. Pitas, and N. Nikolaidis, "Robust audio watermarking in the time domain," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 232–241, Jun. 2001.

[23] Y. Xiang, I. Natgunanathan, Y. Rong, and S. Guo, "Spread spectrum-based high embedding capacity watermarking method for audio signals," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 12, pp. 2228–2237, Dec. 2015.

[24] Y. Xiang, I. Natgunanathan, D. Peng, W. Zhou, and S. Yu, "A dual-channel time-spread echo method for audio watermarking," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 383–392, Apr. 2012.

[25] G. Hua, J. Goh, and V. L. L. Thing, "Time-spread echo-based audio watermarking with optimized imperceptibility and robustness," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 2, pp. 227–239, Feb. 2015.

[26] N. K. Kalantari, M. A. Akhaee, S. M. Ahadi, and H. Amindavar, "Robust multiplicative patchwork method for audio watermarking," *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 6, pp. 1133–1141, Aug. 2009.

[27] X. He and M. S. Scordilis, "An enhanced psychoacoustic model based on the discrete wavelet packet transform," *J. Franklin Inst.*, vol. 343, no. 7, pp. 738–755, 2006.

[28] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451–515, Apr. 2000.

[29] H. Traunmüller, "Analytical expressions for the tonotopic sensory scale," *J. Acoust. Soc. Amer.*, vol. 88, no. 1, pp. 97–100, 1990.

[30] H.-T. Hu, L.-Y. Hsu, and Y.-H. Chang, "An effective correlation formula for enhancing the detectability of spread spectrum-based watermarking," in *Proc. 41st Int. Conf. Telecommun. Signal Process.*, Jul. 2018, pp. 1–5.

[31] C.-M. Pun and X.-C. Yuan, "Robust segments detector for desynchronization resilient audio watermarking," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 11, pp. 2412–2424, Nov. 2013.

[32] M. Goto, "Development of the RWC music database," in *Proc. 18th Int. Congr. Acoust. (ICA)*, 2004, pp. I-553–I-556.

[33] P. Kabal, "An examination and interpretation of ITU-R BS.1387: Perceptual evaluation of audio quality," Dept. Elect. Comput. Eng., McGill University, TSP Lab, Tech. Rep., 2002. Accessed: Jan. 23, 2019. [Online]. Available: http://www-mmsp.ece.mcgill.ca/Documents/Reports/

[34] S. Xiang, "Audio watermarking robust against D/A and A/D conversions," *EURASIP J. Adv. Signal Process.*, vol. 2011, no. 3, pp. 1–14, 2011. doi: 10.1186/1687-6180-2011-3.

**HWAI-TSU HU** received the B.S. degree from National Cheng Kung University, Taiwan, in 1985, and the M.S. and Ph.D. degrees from the University of Florida, USA, in 1990 and 1993, respectively, all in electrical engineering. Since 1998, he has been a Professor with the Department of Electronic Engineering, National I-Lan University, Taiwan. His research interests include speech, audio, and image signal processing.

**TUNG-TSUN LEE** received the B.S. and M.S. degrees in computer science and engineering from National Chiao Tung University, Taiwan, in 1983 and 1985, respectively. Since 1992, he has been a Lecturer with the Department of Electronic Engineering, National I-Lan University, Taiwan. His research interests include software engineering and computer networks.