Design and Analysis of a Scalable Watermarking Scheme for the Scalable Audio Coder

Zhi Li, *Student Member, IEEE,* Qibin Sun, *Member, IEEE,* and Yong Lian, *Senior Member, IEEE*

***Abstract*— In this paper, we present a scalable approach to design lossless watermark for audio. The proposed watermarking framework is built on a recently standardized two-layer scalable audio coder AAZ [1]. By embedding watermark in both the core layer and enhancement layer bitstreams in a special way, the watermark distortion in either layer is compensated by the watermark in the opposite layer. The proposed spread-spectrum-based solution overcomes both the problem of introducing non-invertible distortions in lossy watermark approaches and the problem of non-adaptive embedding in lossless watermarking approaches. Theoretic analysis and experiment results further confirm the validity of the proposed framework in terms of payload, robustness, data expansion property and perceptual quality.**

***Index Terms*— watermarking, scalable audio coding, spread spectrum, AAZ**

I. Introduction

Digital watermarking techniques have been studied for several years for various types of media content such as image, audio and video [2]–[22]. Depending on applications, digital watermarking can be mainly categorized into fragile (or semi-fragile) watermarking for authentication and robust watermarking for copyright protection, content annotation and *etc*. Based on the unique features of audio signal and human auditory system (HAS), various audio watermarking techniques have been proposed [3], [4], [8], [13], [15], [18], [19], [23]. For example, in [3], the proposed echo hiding technique makes use of the temporal masking effect of human ears. In [2], the information is embedded by modulating the phase of the audio signal. In [18], the watermark is represented by sinusoidal patterns and embedded to the host audio. However, the mainstream of audio watermarking is spread-spectrum (SS) based watermarking [4], [13], [15], [19].

A. Motivation and Approaches

Most of the traditional watermark methods protect the media content in a lossy way, *i.e.*, once the watermark is embedded into the media content, the distortion introduced is non-invertible. While this distortion is relatively small and inaudible for most of the average users, it is not suitable for applications such as audio archiving, studio, high-quality streaming and high-end consumer electronic applications, which usually have lossless compression requirement. Further applying a lossy watermarking scheme would render the lossless compression meaningless. These applications motivate to develop lossless watermark in which the original content is still recoverable after watermark embedding. Some lossless watermark methods have been proposed for images [12], [20], [22]. However, one common shortness of these approaches is that the watermark strength has to be independent of the local gray-level values in order to make it invertible. While this is an acceptable requirement for image, it may not be applicable for audio, because the HAS is much more sensitive than the human visual system (HVS). Therefore, in order to be more imperceptible, the embedded watermark has to be adaptive to the host signal. Besides, the methods in [12], [22] typically bind the watermark in the insignificant components of the media content (*e.g.*, bit-planes of smooth area in an image); as a result, the watermark is not robust.

To circumvent these problems, in this paper, an alternative approach – scalable watermarking which builds a bridge between lossy and lossless watermarking – is presented. Like other lossy watermark schemes, the scalable watermarking scheme binds the watermarks with the most significant components of the content so that if one wants to destroy the embedded watermarks one may also have to seriously destroy the content to be protected. In the meantime, it also owns the nice property from lossless schemes – the original content can be exactly recovered from the uncorrupt watermarked content. The proposed framework is based on the Advanced Audio Zip (AAZ) coder [1], which has been adopted in the Final Draft of International Standard (FDIS) for the on-going scalable audio coding standard under MPEG4 [24]. We therefore call our system AAZ-WM. The AAZ coder is a two-layer scalable audio coder, in which the core layer is backward compatible with the well-known AAC coder [25] whereas the enhancement layer (LLE) is an embedded entropy coder, named Bit-Plane Golomb Code (BPGC) [26], serving the transcoding purpose. The AAZ-WM system is fully incorporated into the AAZ coder, binding watermarks to both layers. The watermarks are designed in such a way that the watermark distortion in either layer is compensated by the watermark in the opposite layer. When merging the two layers at the decoder, the watermarks cancel out and the original lossless audio is recovered.

The underlying algorithm is SS-based watermarking. One great feature of this algorithm is its robustness. Of course, watermarking is like a competitive game between designers and adversaries, and the basic SS-based algorithm is not sufficient to defeat all kinds of tricky attacks. Intensive research work has been done to improve the basic SS-based algorithm to combat various attacks. In [15], Malvar and Kirovski have done some excellent work on improving the robustness of ba-

Z. Li and Y. Lian are with the Department of Electrical and Computer Engineering, National University of Singapore (e-mail: lizhi@nus.edu.sg; eleliany@nus.edu.sg). Q. Sun is with the Institute for Infocomm Research, A-STAR (e-mail: qibin@i2r.a-star.edu.sg).

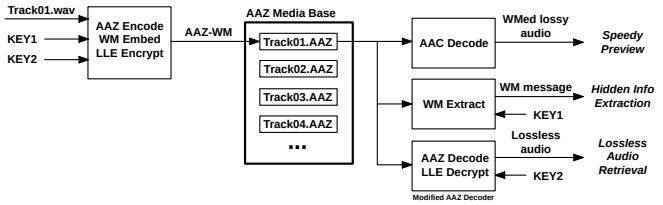

Fig. 1. AAZ-WM application scenario of lossless audio archiving. The raw audio is encoded into AAZ-WM format and stored in the AAZ media base. The system facilitates speedy preview, secure hidden information extraction and secure lossless audio retrieval.

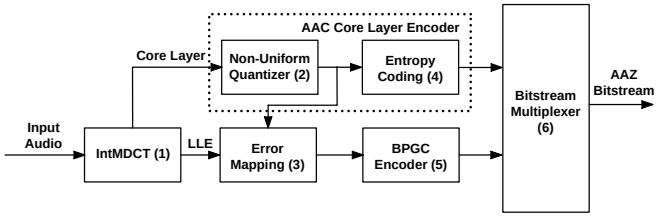

Fig. 2. Structure of the AAZ encoder. AAZ is a two-layer coder where the first layer is backward-compatible with the well-known AAC coder and the second layer compensates the coding loss of the first layer.

sic SS-based algorithm against attacks such as desynchronization, estimation, removal and *etc*. In [23], Tachibana *et al*. have derived algorithms against time and frequency fluctuation. Their contributions have proven the feasibility of applying SS-based algorithm to audio. In this paper, our main objective is to provide a framework of lossless audio watermarking from the system level (instead of algorithm level). The advantage is that many available SS-based algorithms can be directly apply to this system to improve robustness performance. For example, the existing algorithms [15], [23] to defeat desynchronization and watermark estimation can be implemented on top of the proposed system.

Under this framework, the embedded watermark can be made well adaptive to the local host audio. Three techniques - i) HAS-based perceptual shaping ii) host signal compensation and iii) adaptive watermark allocation - are presented, which "tailor" the watermark to the local strength to reduce the watermark impact on the audio quality.

Note that as a feature of lossless watermarking, once the audio is losslessly recovered, the watermark is fully removed. In view of its potential security issue, in this work we have implemented an encryption-based approach to restrict unauthorized watermark removal and control the access to the lossless audio. The LLE layer bitstream is encrypted using a secret key (KEY2) before multiplexed with the core layer bitstream (refer to Figure 4). Some related work on JPEG2000 secure transcoding has been made in [27]. Throughout this paper, we shall not elaborate this issue further since it is out of the main scope of this paper.

The proposed AAZ-WM system meets many applications which have lossless audio quality requirement. One application for lossless audio archiving is demonstrated in Figure 1. In records company or studio, thousands of tracks needs to be stored in the lossless format. In the meanwhile, watermark is preferred to embed information such as unique identification numbers, since it provides resilience to malicious information alteration. In this scenario, AAZ-WM provides a good solution. The raw audios are encoded into AAZ-WM format and then stored in the AAZ media base. When browsing for a particular track, low-quality audio for speedy preview is generated by decoding the core layer bitstream only. For authorized users with KEY1, the hidden information can be extracted. For lossless audio retrieval, the authorized user can use KEY2 to decrypt the LLE layer, thereby removing the watermark and recovering the lossless audio. However, without KEY2, audio signal with lossy but acceptable quality can still be generated. AAZ-WM's another great feature is that the watermark is "scalable" in the sense that the watermark strength varies as the transcoding rate varies – when the LLE layer is fully transcoded, the watermark strength in the final decoded audio signal is the strongest; when there is no transcoding and the LLE layer fully compensates the core layer, the watermark is gone. This feature motivates an innovative application – since the watermark strength is an indictor of the transcoding rate, we could use the watermark to "blindly" (*i.e.*, in the absence of the original audio signal) assess the audio quality.

B. Organization of the Paper

The remaining part of this paper is organized as follows. In Section II, background of the scalable audio coder AAZ is briefly introduced. In Section III, a generic model of the watermark system for layered scalable coders is presented. Theoretical analysis of system performance is conducted thereafter. In Section IV, the complete AAZ-WM framework is described while some practical implementation issues are addressed. Experiment results which evaluate the system performance in terms of payload, robustness, data expansion and perceptual quality are given in Section V, followed by conclusions in Section VI.

II. Background

A. Advanced Audio Zip (AAZ) Coder

Scalable audio coding is the technique of encoding the audio bitstream in a convenient way such that the output bitrate can be arbitrarily controlled according to some requirements. A standardized scalable audio coder is the AAZ coder. For the full reference, please see [1], [24].

Refer to Figure 2. The AAZ encoder consists of two layers – the core layer which is essentially an AAC encoder [25] and the LLE layer. First of all, the time-domain audio signal (in PCM format) is losslessly transformed into frequency-domain coefficients by using integer Modified Discrete Cosine Transform (intMDCT) (Module 1). Denote the frequency domain coefficients $\mathbf{c} = [c(1), c(2), ..., c(K)]^T$, where 1024 elements of $c(k)$ form one intMDCT block. K is the number of coefficients used for watermark embedding for one bit of message. Each intMDCT block is further divided into a number of scale-factor bands, each having an optimized

scale-factor calculated from the "quantization and coding" process [25]. The scale-factor of the band where $c(k)$ belongs to is denoted $SF[c(k)]$. $c(k)$ is then passed to the AAC core layer encoder, where quantized by an non-uniform quantizer $Q(\cdot)$ (Module 2):

$$\begin{aligned} Q[c(k)] &= i(k) \\ &= int[(\frac{|\alpha c(k)|}{\sqrt[4]{2^{SF[c(k)]}}})^{3/4} + 0.4054]sgn[c(k)] \end{aligned} \quad (1)$$

where $int(\cdot)$ is the integer operator and $sgn(\cdot)$ is the sign operator. α is a isotropic factor used in order to approximate the outputs of the MDCT filterbank used in AAC [25]. $\mathbf{i} = [i(1), i(2), ..., i(K)]^T$ is the quantized intMDCT (QintMDCT) coefficients. Next, $i(k)$ is further Huffman-coded (Module 4) to produce the core layer bitstream.

In the LLE layer, the output of the non-uniform quantizer $i(k)$ is fed to the error-mapping process (Module 3), where the quantization threshold $thr[i(k)]$ is determined by:

$$\begin{aligned} &Q^{-1}[i(k)] = thr[i(k)] \\ &= \begin{cases} int[\dfrac{\sqrt[4]{2^{SF[c(k)]}}(|i(k)| - 0.4054)^{4/3}}{\alpha}]sgn[i(k)] & , i(k) \neq 0 \\ 0 & , i(k) = 0 \end{cases} \end{aligned} \quad (2)$$

$thr[i(k)]$ is then subtracted from $c(k)$, to produce the residue $e(k)$, $\mathbf{e} = [e(1), e(2), ..., e(K)]^T$. $e(k)$ is further BPGC-encoded in the BPGC encoder (Module 5) to generate the LLE bitstream. In order to facilitate transcoding, $e(k)$ is bit-plane coded progressively from the MSB plane to the LSB plane. In the final stage, the core layer and LLE layer bitstreams are multiplexed to generate the final AAZ bitstream (Module 6).

III. A Generic System Model

In this section, we introduce the generic model of the proposed watermark system. This model is designed based on the AAZ coder, but is applicable to any two-layer-structured scalable coder. More importantly, it facilitates conducting theoretical analysis on the system performance, which is presented in Section III-B.

A. Descriptions

Refer to Figure 3. In the original layered scalable coder (a), the signal $\mathbf{x}$ is firstly transformed into the frequency domain coefficients $\mathbf{c}$ via $T(\cdot)$ operation. For achieving coding gain, the coefficients $c(k)$ are then quantized by $Q(\cdot)$ before being entropy coded in Layer I. Therefore, $i(k)$, element of $\mathbf{i} = [i(1), i(2), ..., i(K)]^T$ is given by:

$$i(k) = Q[c(k)] = Q[T(x(k))] \quad (3)$$

In Layer II, the residue $e(k)$ of $\mathbf{e} = [e(1), e(2), ..., e(K)]^T$ is given by

$$e(k) = c(k) - thr[i(k)] = c(k) - Q^{-1}[Q(c(k))] \quad (4)$$

where $Q^{-1}(\cdot)$ is the inverse quantizer. The residue $e(k)$ is used to compensate the quantization distortion introduced by $Q(\cdot)$.

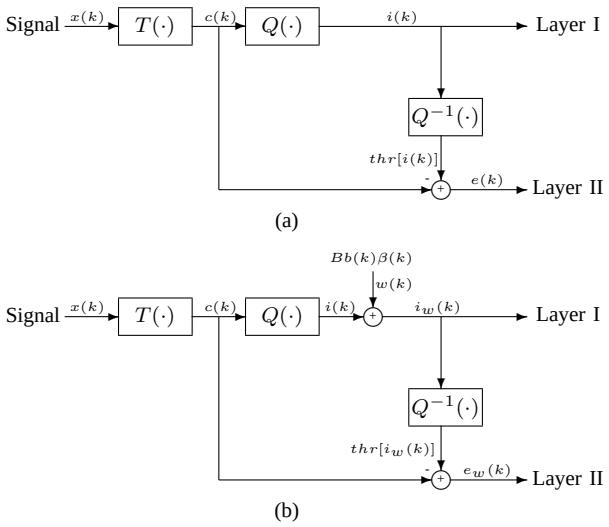

Fig. 3. Generic model of (a) the layered scalable coder and (b) the watermark embedder for layered scalable coder. The watermark signal is added to the Layer I bitstream before it is inverse quantized and passed to Layer II, leading to "automatic" watermarking of the Layer II bitstream.

In the watermark embedder (Figure 3(b)), the watermark is embedded to the quantized coefficients in Layer I. We implement a SS-based watermarking approach. The watermark message bit B is firstly spread by a spreading sequence $\mathbf{b}$, which has K chips and is generated from a secret key. The spread signal is then perceptually shaped by the local watermark strength β, where $\beta(k) \geq 0$. Since $i(k)$ is integer, $\beta(k)$ has to be integer as well. The watermark is $w(k) = Bb(k)\beta(k)$. The watermarked quantized coefficients are:

$$i_w(k) = i(k) + w(k) = i(k) + Bb(k)\beta(k) \quad (5)$$

Now the residue element $e_w(k)$ of $\mathbf{e}_w = [e_w(1), e_w(2), ..., e_w(K)]^T$ is then given by:

$$e_w(k) = c(k) - Q^{-1}[i_w(k)] \quad (6)$$

Therefore we can see that $e_w(k)$ has also been "automatically" watermarked. The extraction of message bit in Layer I is done by correlating the received coefficients and the spreading sequence:

$$\chi_1 = < \mathbf{i}', \mathbf{b} > \quad (7)$$

where $< \mathbf{a}, \mathbf{b} >= (1/K)\mathbf{a}^T\mathbf{b}$ is the normalized inner product. $\mathbf{i}' = [i'(1), i'(2), ..., i'(K)]^T$ is the noisy quantized coefficient received in the extractor. χ_1, the decision statistic, is also a good measure of the watermark strength. The estimated bit is:

$$\hat{B_1} = sgn(\chi_1) \quad (8)$$

Similarly, for Layer II, the extraction criteria is

$$\chi_2 = < \mathbf{e}', \mathbf{b} > \quad (9)$$

$$\hat{B_2} = -sgn(\chi_2) \quad (10)$$

where $\mathbf{e}' = [e'(1), e'(2), ..., e'(K)]^T$ is the noisy residue. In order to recover the original lossless signal, we need to

perfectly reconstruct the frequency domain coefficients $\mathbf{c}$. In the decoder, this is simply done by:

$$c(k) = e_w(k) + Q^{-1}[i_w(k)] \tag{11}$$

B. Theoretical Analysis of System Performance

For simplicity, we make the following assumptions for analysis:

- The quantizer and inverse quantizer are linear, given by:

$$Q[c(k)] = int[c(k)/\lambda] \tag{12}$$

$$Q^{-1}[i(k)] = \lambda i(k) \tag{13}$$

  where λ is the quantization step size.
- The perceptual shaping β is constant, denoted by β_c.
- The additive noise $\mathbf{n}_1 = [n_1(1), n_1(2), ..., n_1(K)]^T$ (due to attack, compression and *etc*.) to the transform coefficients is Gaussian. In addition, the compensation signal from Layer II is $\mathbf{n}_e(\hat{R}_{trc}) = [n_e(1, \hat{R}_{trc}), n_e(2, \hat{R}_{trc}), ..., n_e(K, \hat{R}_{trc})]^T$, where $\hat{R}_{trc}$ is the normalized transcoding rate (refer to Appendix I). Note that $\mathbf{n}_e(\hat{R}_{trc})$ is present in Extraction Scenario 1 only (see Section IV-B.1). The received noisy signal is

$$i(k)' = i_w(k) + n_e(k, \hat{R}_{trc}) + n_1(k) \tag{14}$$

- Similarly, the additive noise $\mathbf{n}_2 = [n_2(1), n_2(2), ..., n_2(K)]^T$ to the residue in layer II is Gaussian. The received signal is

$$e(k)' = e_w(k) + n_2(k) \tag{15}$$

In Appendix I, we show that in Layer I, the watermarked quantized coefficient $i_w(k)$ plus the compensation signal $n_e(k, \hat{R}_{trc})$ from Layer II can be expressed as:

$$i_w(k) + n_e(k, \hat{R}_{trc}) = i(k) + B\beta_c b(k) 2^{\hat{R}_{trc} - L_e} \tag{16}$$

where L_e is the number of bits to represent $e_w(k)$ in binary form. Therefore, for Layer I, from Equation 7, 14 and 16,

$$\chi_1 = B\beta_c 2^{\hat{R}_{trc} - L_e} + \frac{1}{K}\mathbf{i}^T\mathbf{b} + \frac{1}{K}\mathbf{n}_1^T\mathbf{b} \tag{17}$$

Assume that $c(k)$ follows a Gaussian Distribution of zero mean and variance σ_c^2, *i.e.*, $c(k) \sim N(0, \sigma_c^2)$, thus $i(k) \sim N(0, \sigma_c^2/\lambda^2)$. Also assume $n_1(k) \sim N(0, \sigma_{n1}^2)$. Hence, $(1/K)\mathbf{i}^T\mathbf{b} \sim N[0, \sigma_c^2/(\lambda^2 K)]$ and $(1/K)\mathbf{n}_1^T\mathbf{b} \sim N(0, \sigma_{n1}^2/K)$. The bit error rate (BER) is given by:

$$\begin{aligned} P_{e,1}(\hat{R}_{trc}) &= P(\chi_1 < 0 | B = 1) \\ &= \frac{1}{2}\text{erfc}\left(\sqrt{\frac{\beta_c^2 2^{2(\hat{R}_{trc} - L_e)} K}{2\sigma_c^2/\lambda^2 + 2\sigma_{n1}^2}}\right) \end{aligned} \tag{18}$$

where $\text{erfc}(x)$ is the complementary error function, defined as

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty exp(-u^2) du \tag{19}$$

For Layer II, let $\mathbf{\Delta} = [\Delta(1), \Delta(2), ..., \Delta(K)]^T$ denote the difference between the watermarked and unwatermarked residue. We have:

$$\begin{aligned} \Delta(k) &= e_w(k) - e(k) \\ &= [c(k) - Q^{-1}(i_w(k))] - [c(k) - Q^{-1}(i(k))] \\ &= -\lambda B \beta_c b(k) \end{aligned} \tag{20}$$

Therefore,

$$e_w(k) = e(k) - \lambda B \beta_c b(k) \tag{21}$$

From Equation 9, 15 and 21,

$$\chi_2 = \frac{1}{K}\mathbf{e}^T\mathbf{b} + \frac{1}{K}\mathbf{n}_2^T\mathbf{b} - \beta_c \lambda B \tag{22}$$

$e(k)$ follows a uniform distribution between $-\lambda$ and λ (consider $e(k)$ as LSBs of $c(k)$). Therefore, $e(k)$ has zero mean and variance $\lambda^2/3$. Using Central Limit Theorem, we therefore have $(1/K)\mathbf{e}^T\mathbf{b} \sim N(0, \lambda^2/3K)$. Assume $n_2(k) \sim N(0, \sigma_{n2}^2)$, therefore $(1/K)\mathbf{n}_2^T\mathbf{b} \sim N(0, \sigma_{n2}^2/K)$. The BER is given by

$$P_{e,2} = P(\chi_2 > 0 | B = 1) = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{\beta_c^2 K}{2/3 + 2\sigma_{n2}^2/\lambda^2}}\right) \tag{23}$$

Note that robustness, fidelity and data payload are three mutually contradictory requirements for a watermark scheme. This can be evidenced from Equation 18 and 23. Robustness is measured by watermark extraction BER P_e whereas fidelity is measured by β_c. for a given P_e, the allowable data payload (*i.e.*, watermark message rate) is

$$R_1 = \frac{r_{cs} R_0 \beta_c^2 2^{2(\hat{R}_{trc} - L_e)}}{[\text{erfc}^{-1}(2P_e)]^2 (2\sigma_c^2/\lambda^2 + 2\sigma_{n1}^2)} \tag{24}$$

for Layer I, and

$$R_2 = \frac{r_{cs} R_0 \beta_c^2}{[\text{erfc}^{-1}(2P_e)]^2 (2/3 + 2\sigma_{n2}^2/\lambda^2)} \tag{25}$$

for Layer II, where r_{cs} is the coefficient selection rate, *i.e.*, percentage of coefficients selected for watermark embedding, and R_0 is the bitstream sampling rate. It is evidenced that R_1 is less than R_2. Intuitively, this is because that the core layer host signal power is stronger than the LLE layer residue signal power. Therefore, the system data payload is upper-bounded by R_1.

IV. The AAZ-WM System

In this section, we describe the proposed AAZ-WM framework in detail. Some of the practical issues regarding implementing this framework in the AAZ coder are addressed. In Section IV-A, the AAZ encoder / AAZ-WM embedder is presented. Three techniques for improving the watermark adaptiveness, namely i) HAS-based perceptual shaping, ii) host signal compensation and iii) adaptive watermark allocation are presented in Section IV-A.1 to IV-A.3. Section IV-B describes the AAZ-WM extractor. Three different watermark extraction scenarios are detailed in Section IV-B.1, followed by an algorithm for detecting the presence of watermark in Section IV-B.2. Section IV-C presents a brief complexity analysis of the AAZ-WM system.

A. AAZ Encoder / AAZ-WM Embedder

The complete structure of the AAZ encoder / AAZ-WM embedder is illustrated in Figure 4. The shaded blocks illustrate the embedded AAZ-WM modules. Their functions will be detailed in the next several sections.

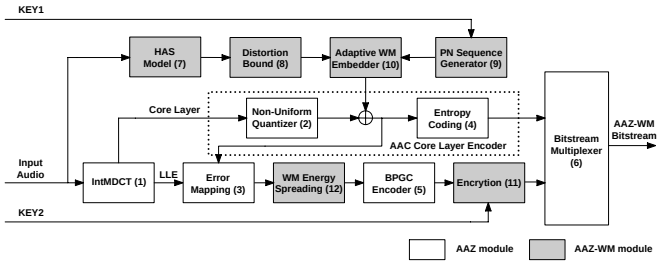

Fig. 4. Structure of the proposed AAZ encoder / AAZ-WM embedder. The unshaded blocks illustrate the original AAZ modules and the shaded blocks illustrate the embedded AAZ-WM modules.

Refer to Figure 4, the watermark $\mathbf{w}$, generated from Module 10, is embedded to the quantized intMDCT coefficients $\mathbf{i}$ after the non-uniform quantization (Module 2) in the core layer, such that the embedded watermark survives the quantization process "by nature". Besides, the watermark embedding takes place before $\mathbf{i}$ is fed to the error-mapping process (Module 3). Consequently, the reconstructed intMDCT coefficient is modified by the watermark, and therefore the residue $\mathbf{e}$ is also modified. In other words, the LLE layer bitstream is "automatically" watermarked.

The watermark is chosen to be added to the perceptually significant bands (*i.e.*, the near-DC components) of the spectrum so that it cannot be illegally removed without sacrificing the fidelity. Increasing the embedding bandwidth (thereby increasing the coefficient selection rate r_{cs}, refer to Section III-B) will give more space for watermark, leading to increased payload. However, the price to pay is that the file size (or data rate) will expand accordingly. In system design one needs to consider the trade-off of these two factors by choosing a proper embedding bandwidth.

Note that in Section III, we introduced the extraction criteria as in Equation 7, 8 and Equation 9, 10. It is not difficult to prove these extraction criteria are applicable if $Q(\cdot)$ and $Q^{-1}(\cdot)$ are linear quantizers. The non-uniform quantizer (Equation 1 and 2) used in AAZ actually has very similar property as a linear quantizer, and thus the same extraction criteria can be applied as well. We will provide experiment results to illustrate this issue in Section V. Mathematical proof of this property will be given in Appendix I.

Module 11 is employed to control the access to lossless audio stream, as introduced in Section I-A.

1) HAS-based Perceptual Shaping: In order to enhance the audio fidelity, a HAS perceptual model (Module 7) is utilized to compute the bound of distortions unperceptual by human ears. This allowable distortion bound is output in terms of signal-to-masking ratio (SMR) for each intMDCT coefficient. The SMR for coefficient $c(k)$ is denoted by $SMR[c(k)]$.

In Module 8, firstly, the QintMDCT coefficient $i(k)$ is reconstructed using Equation 2. For a scale factor band of indices $k_1 \le k \le k_2$, the total energy is $\sum_{k=k_1}^{k_2} c(k)^2$, therefore the total allowable distortion is $[\sum_{k=k_1}^{k_2} c(k)^2]/SMR[c(k)]$. For the allowable distortion for each coefficient, it is desirable to make it proportional to the coefficient value, *i.e.*, the distortion is $\delta c(k)$, where δ is a constant. Therefore, the total allowable distortion within a scale-factor band is:

$$\sum_{k=k_1}^{k_2} [\delta c(k)]^2 = [\sum_{k=k_1}^{k_2} c(k)^2]/SMR[c(k)] \tag{26}$$

Therefore,

$$\delta = 1/\sqrt{SMR[c(k)]} \tag{27}$$

Hence, the distortion bounds are given by:

$$\begin{aligned} c_-(k) &= Q^{-1}[i(k)](1-\varepsilon/\sqrt{SMR[c(k)]}) \\ c_+(k) &= Q^{-1}[i(k)](1+\varepsilon/\sqrt{SMR[c(k)]}) \end{aligned} \tag{28}$$

where ε is a global strength bound. Here we introduce ε in order to allow some flexibility for distortion control. Ideally, when $\varepsilon = 1$, the distortion is just masked by the host signal, and is controlled within the range of just noticeable distortion (JND). However, due to the HAS model limitations, some distortions are still audiable. That is why we need some further improvement on watermark adaptiveness in the later sections. Besides, for low bitrate where the major distortion is due to quantization error, ε can be set to larger than 1. The distortion bounds are then converted to the bounds for the QintMDCT coefficient:

$$\begin{aligned} i_-(k) &= Q[c_-(k)] \\ i_+(k) &= Q[c_+(k)] \end{aligned} \tag{29}$$

Therefore, $i_w(k)$ (*i.e.*, the output of Module 8) is bounded by:

$$i(k) + \beta_-(k) \le i_w(k) \le i(k) + \beta_+(k) \tag{30}$$

$$i_-(k) \le i_w(k) \le i_+(k) \tag{31}$$

2) Host Signal Compensation: In this section, we further improve the watermark adaptiveness by compensating the host signal influence during embedding. In the literature, some researchers have made the proposal that since we have the perfect knowledge of the host signal at the watermark embedder, we could model the watermarking problem as communications with side information [9], [17], [28]. In this paper, we demonstrate Malvar *et al.*'s Improved Spread Spectrum (ISS) algorithm in practical use. In [17], Malvar *et al.* give theoretical analysis on improving the robustness of SS-based algorithm and proposed a new technique called ISS. However, one practical issue is that, when the distortion needs to be locally bounded, the performance is dramatically reduced. Nevertheless, ISS provides a good method for controlling the watermark strength. In this paper, we provide an alternative use of ISS - fine-tuning the watermark distortion to improve the audio signal fidelity. We also demonstrate how ISS facilitates the watermark presence detection, which is presented in Section IV-B.2.

The rationale of host signal compensation is, since in the embedder, we have the complete knowledge of the host signal, we can effectively calculate the correlation between the host signal and the spreading sequence, estimate its impact on the watermark extraction, and therefore determine the watermark strength needed to maintain a fixed level of robustness.

In the core layer, the robustness is measured by χ_1 and the host signal is $\mathbf{i}$. Assume the message bit B is 1, and we want

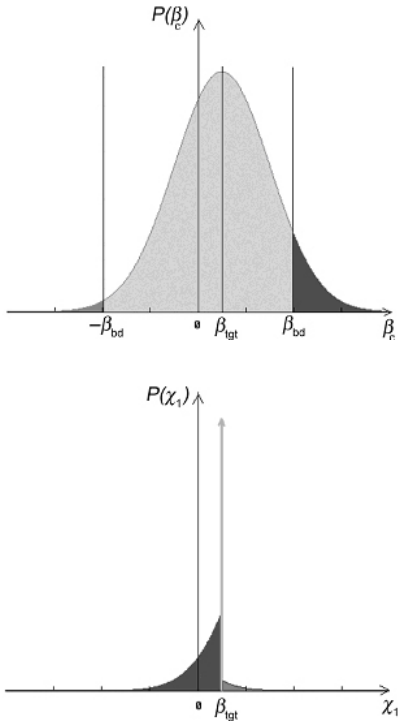

Fig. 5. Expected distribution of compensated watermark strength β_c and extraction statistic χ_1. A Guassian model is used.

to maintain the robustness at a target level β_{tgt}. Here we also ignore the external attack $\mathbf{n_1}$. From Equation 17, we have:

$$\chi_1 = \beta_c + \frac{1}{K}\mathbf{i}^T\mathbf{b} = \beta_{tgt} \tag{32}$$

Therefore, the total amount of watermark to embed in one watermark bit duration is:

$$W_{bit} = K\beta_c = K\beta_{tgt} - \mathbf{i}^T\mathbf{b} \tag{33}$$

Note that each QintMDCT coefficient is bounded by Equation 30, where the bound is locally adaptive. For analysis, a constant bound β_{bd} is assumed. Suppose W_{bit} is bounded by:

$$-K\beta_{bd} \le W_{bit} \le K\beta_{bd} \tag{34}$$

W_{bit} is clipped at $\pm K\beta_{bd}$ if it exceeds the bound. Figure 5 illustrates the expected probability density function (p.d.f.) of β_c and χ_1 based on the analysis above. The extraction error occurs when χ_1 is less than 0. With the presence of attack $\mathbf{n}_1$, the BER is

$$P'_{e,1} = P(\chi_1 < 0 | B = 1) = \frac{1}{2}\text{erfc}(\sqrt{\frac{\beta_{bd}^2 K}{2\sigma_c^2/\lambda^2 + 2\sigma_{n1}^2}}) \tag{35}$$

Comparing Equation 35 with Equation 18 (consider the best case when the LLE layer is fully truncated, *i.e.*, $\hat{R}_{trc} = L_e$), we can interpret that the BER of watermark extraction has been maintained at the same level if we let $\beta_{bd} = \beta_c$. However, since the average watermark value is smaller than β_{bd}, the original audio signal is now less distorted by the watermark signal.

TABLE I

FIDELITY IMPROVEMENT AFTER HOST SIGNAL COMPENSATION AND ADAPTIVE WATERMARK ALLOCATION

Test Seq.	es01	es02	es03	sc01	sc02	sc03
SNR (before)	18.556	21.135	21.301	22.785	20.094	14.153
SNR (after)	21.048	24.565	25.833	28.897	26.146	19.851
ODG (before)	-0.91	-0.84	-1.28	-2.45	-0.88	-0.57
ODG (after)	-0.60	-0.76	-1.03	-1.77	-0.81	-0.50
Test Seq.	si01	si02	si03	sm01	sm02	sm03
SNR (before)	15.232	14.889	23.392	27.258	32.424	19.000
SNR (after)	18.405	15.306	29.109	30.510	34.853	24.231
ODG (before)	-1.11	-2.21	-1.83	-1.53	-3.89	-0.56
ODG (after)	-0.69	-1.99	-1.18	-0.57	-3.36	-0.53

(Note: The improvement is measured in terms of SNR and ODG. For test details of ODG, please refer to Section V-D.)

3) Adaptive Watermark Allocation: Now let us look at how to adaptively allocate the watermark to the QintMDCT coefficients in the core layer (Module 10). In the previous analysis we assume the watermark strength β_c is constant (refer to Section III-B). In practice, however, we can embed the watermark at will, as long as the overall amount is W_{bit} for one watermark bit duration.

In Module 9, a spreading pseudo-random (PN) sequence is generated according to KEY1. KEY1 fully determines the secrecy of the hidden information (Module 9). The computed watermark is to be adaptively allocated with reference to the spreading sequence, which determines the polarity (*i.e.*, positive or negative) of the watermark. The aim now is to "fill" the watermark adaptively to the local host signal such that the watermark would have least impact on the audio fidelity. Let us define "watermark capacitance" as the maximum amount of watermark allowed for each QintMDCT coefficient, as bounded by Equation 30. The adaptive watermark allocation strategy in the core layer is based on the following criterion: the watermark embedded in the coefficient with higher watermark capacitance is less audiable than others, thus have higher priority for watermark allocation. Based on this rule, the watermark is allocated iteratively. The following pseudo code demonstrates this procedure:

```
//compute watermark capacitance
for each i(k)
  compute wm_cap(i(k))
//iterative watermark allocation
for w = 1 to W_bit
  find i_emb of i(k) which has maximum wm_cap(i(k))
  embed one bit watermark to i_emb
  wm_cap(i_emb) = wm_cap(i_emb) − 1
  if for all i(k), wm_cap(i(k)) == 0, then terminate
```

Table I illustrates the fidelity improvement for the 12 test sequences after host signal compensation and core-layer adaptive watermark allocation is implemented (for details of the 12 sequences of test audio, please refer to Section V).

In the LLE layer, it is noticed that the watermarked residue $e_w(k)$ displays significant magnitude increase. This non-adaptiveness will cause inefficiency in the subsequent BPGC encoding (Module 5), as well as potential security issues. Therefore, the watermark adaptiveness in the LLE layer must also be improved. As a solution, we spread the watermark energy to a wider interval in the frequency domain (Module 12). Originally, the watermark is only embedded to the first dozens of residues while much space in the rest of the spectrum is available to accommodate the watermark energy. Assume that each coefficient has β_{expd} bit-planes expanded

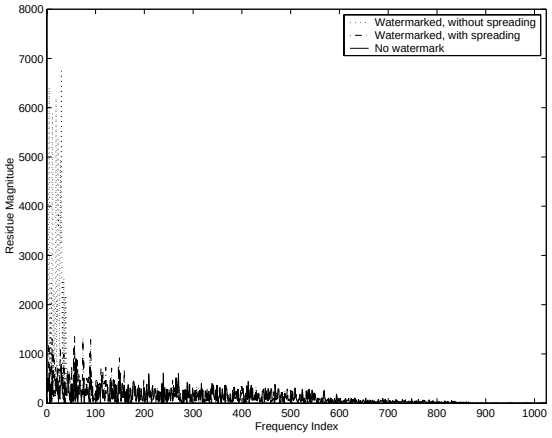

Fig. 6. Effect of watermarked residue energy spreading. After spreading the watermark energy from coefficients indexed from 0 - 39 to 0 - 159, the residue distribution is much more closer to the original distribution. Further improvement can be achieved by spreading the energy to even wider range in the spectrum.

maximally (we have implemented algorithms to control the maximum number of bit-plane expansion). Now we consider to spread the β_{expd} bits for each watermarked residue to β_{expd} different residues, each having one expanded bit-plane. In order to make the watermark robust, we must ensure that they are the last bits to be truncated in the transcoding process; therefore, the additional bit-plane is placed on top of the MSB bit-plane of the original residue in the perceptually significant bands [1]. This will not cause significant impact on the audio fidelity, since residues are perceptually less important. Figure 6 demonstrates the spreading effect to the residue magnitudes. More results are presented in Section V-E.

B. Watermark Extraction

The extraction of the watermark message bits could be performed in three scenarios, including: i) message extraction from AAZ-decoded PCM audio signal, ii) message extraction from the core layer of the AAZ bitstream, iii) message extraction from the LLE layer of the AAZ bitstream.

However, in all three scenarios, the extraction is "mechanical" in the sense that no matter whether the audio signal has been watermarked or not, some message bits would be extracted anyway. Therefore, we need an additional step to detect the presence of watermark signal, in order to enhance the confidence level of extraction. Some approaches have been proposed for watermark detection of high-payload watermark in literature [29] and [14]. One common approach is to test each bit independently, and report that the watermark is present only if every bit's decision statistic exceeds a threshold [29]. Obviously, this approach may lead to very poor false negative rate. We argue that the watermark presence is better determined jointly by the extraction statistics (*i.e.*, χ_1 for the core layer and χ_2 for the LLE layer) for all watermark bits computed during the extraction process. After detection, if the watermark is declared present, the extracted message bits are output; otherwise, the extracted bits are discarded. Details will be given in Section IV-B.2.

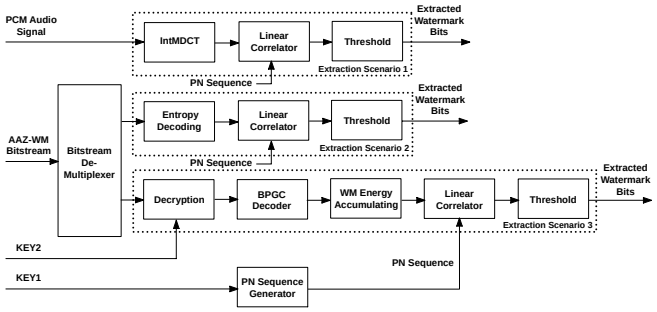

Fig. 7. Watermark message extraction i) from AAZ-decoded PCM audio signal ii) from the core layer of the AAZ bitstream iii) from the LLE layer of the AAZ bitstream. In all scenarios, the PN sequence is generated from KEY1.

1) Three Extraction Scenarios: Figure 7 illustrates the three watermark extraction scenarios. In one case, we want to extract watermark from a decoded PCM audio signal (Scenario 1). Note that not in all cases, the watermark is present in the PCM audio. For example, when the AAZ-WM bitstream is losslessly decoded, the watermark is gone; when the LLE layer transcoding rate is low, the watermark is essentially gone. However, when the AAZ-WM bitstream is decoded using a AAC decoder (refer to Section II), or when decoded by an unauthorized user who does not have KEY2, the watermark is preserved in the PCM audio. To extract the watermark, similar to in the AAZ encoder, the PCM audio is firstly intMDCT transformed. The intMDCT coefficient is then correlated with the spreading PN sequence generated from KEY1, and is further thresholded to estimate the message bit.

In other cases, we wish to extract the watermark directly from the encoded AAZ-WM bitstream. This can be performed in the core layer (Scenario 2) or the LLE layer (Scenario 3). The AAZ-WM bitstream is firstly de-multiplexed. In Scenario 2, the core layer bitstream is entropy-decoded to recover $\mathbf{i}_w$. $\mathbf{i}_w$ is then correlated with $\mathbf{b}$ and further thresholded for the estimated watermark bit. In Scenario 3, the LLE layer bitstream must be firstly decrypted using KEY2, followed by BPGC-decoding and watermark energy accumulation (reversion of Module 12 of the AAZ encoder / AAZ-WM embedder). The rest of the steps is similar as above.

In case the watermark strength is to be examined instead of watermark bit extraction, we take the average of the correlation values (χ_1 or χ_2) for all watermark bits as the measurement of watermark strength.

2) Detection of Watermark Presence: In watermark extraction, the sign of each χ_1 determines each extracted bit; in this section, with the help of ISS technique [17], the magnitude $|\chi_1|$ is explored, to reveal the information of watermark presence or absence. The rationale is: if the watermark system is modeled accurately, then the p.d.f. of $|\chi_1|$ can be found in either presence or absence cases. This is done as follows: In Section IV-A.2, the p.d.f. of χ_1 has been modeled (refer to Figure 5), assuming watermark presence. Therefore it is straightforward to model the p.d.f. of $|\chi_1|$. Let us denote the p.d.f. of $|\chi_1|$ as p_1 assuming watermark presence. When the

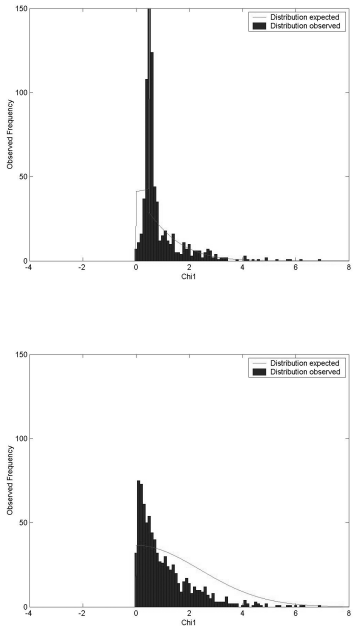

Fig. 8. Expected and observed distribution of the statistic $|\chi_1|$ for watermark presence (left) and absence (right), respectively. Note that β_{tgt} is set to 0.5. A Gaussian model is used for the expected distribution.

watermark is absent, χ_1 is:

$$\chi_1 = \frac{1}{K}\mathbf{i}^T\mathbf{b} \tag{36}$$

Therefore, $\chi_1 \sim N(0, \sigma_i^2/K)$. The p.d.f. of $|\chi_1|$ assuming watermark absence, denoted by p_0, can also be found accordingly. To examine the accuracy of this model, we use χ_1's obtained from Extraction Scenario 2. The observed and expected p.d.f. of $|\chi_1|$ are compared in Figure 8. The observed histogram of $|\chi_1|$ is close to what we have expected in theory, except the distribution is more Laplacian-like. We notice that in practice, it is more accurate to model the distribution of χ_1 as Laplacian instead of Gaussian. This could be one of the future tasks to model the system based on Laplacian distribution. Having modeled the p.d.f., the problem now becomes the following hypothesis testing problem:

$$\begin{cases} H_0 : \text{watermark absent, } |\chi_1| \sim \text{i.i.d. } p_0 \\ H_1 : \text{watermark present, } |\chi_1| \sim \text{i.i.d. } p_1 \end{cases} \tag{37}$$

As the prior probability of watermark presence is unknown, a common solution to this problem is to use ML decision rule. Here we consider an alternative solution. The statistics observed must follow the same p.d.f. as expected in the modeling. We can therefore use Pearson chi-square goodness-of-fit test to examine which distribution in theory the observed data are closer to, thus determine whether the watermark is present in the signal. The detection performance, *i.e.*, the false positive rate and the false negative rate, depends on three factors: i) how accurate we can model the system, ii) the pool size of the available number of χ_1's and iii) the distinguishableness of the two p.d.f.s. For this system, we propose a decision rule which is a simplified version of the goodness-of-fit test. Note that the ISS technique allows fine-tuning of watermark strength and therefore help to distinguish the two p.d.f.s. The decision rule is summarized as follows:

- Obtain the observed statistics χ_1 from the watermark extraction process.
- Calculate the frequency of χ_1 which falls within the region of $(\beta_{tgt}-\beta_\Delta, \beta_{tgt}+\beta_\Delta)$, where β_Δ is the vicinity tolerance. In our experiment, β_{tgt} is set to 0.5 and β_Δ is set to 0.05. Normalize the frequency by the total number of χ_1.
- The decision is made by comparing the normalized frequency to a threshold τ. If the normalized frequency $> \tau$, the watermark is declared present; otherwise, the watermark is declared absent. In our experiment, τ is set to 0.5.

TABLE II
DETECTION OF WATERMARK PRESENCE BASED ON STANDARD TEST SEQUENCES

Sequence		es01	es02	es03	sc01	sc02	sc03
Presence	Norm. Freq.	0.906	0.905	1.000	0.977	1.000	0.987
	Decision	Y	Y	Y	Y	Y	Y
Absence	Norm. Freq.	0.063	0.048	0.000	0.007	0.000	0.000
	Decision	N	N	N	N	N	N
Sequence		si01	si02	si03	sm01	sm02	sm03
Presence	Norm. Freq.	0.969	0.906	0.982	0.952	0.829	0.969
	Decision	Y	Y	Y	Y	Y	Y
Absence	Norm. Freq.	0.031	0.034	0.037	0.024	0.021	0.031
	Decision	N	N	N	N	N	N

The experimental results using the 12 standard test sequences are shown in Table II. The value of the normalized frequency demonstrates the robustness of this approach.

C. Complexity Analysis

Since the watermarking system is incorporated into the AAZ coder, it adds extra complexity to the AAZ coder. We turn to look at how each AAZ-WM module influence the complexity of the overall system. Refer to Figure 4, in the watermark embedder, the main extra complexity comes from the HAS model (Module 7). In the test demo, we have made use of the HAS model which is already incorporated into the core layer AAC coder. Therefore, Module 7 essentially does not contribute extra complexity to the AAZ-WM system. For the other modules, Module 9 only adds some overhead to the embedding procedure; Module 8, 11 and 12 has time complexity of $O(K)$; Module 10 has time complexity of $O(K^2)$ (refer to the pseudo code in Section IV-A.3). The test shows the encoding/embedding time of the AAZ-WM system is slightly longer than the encoding time of the AAZ coder.

The watermark extraction process is similar (Figure 7). The correlation module has complexity of $O(K)$ whereas the threshold module has complexity of $O(1)$. For Extraction Scenario 2 and 3, the extraction can be performed in real-time in the bitstream decoding process. Watermark presence detection only add some overhead to the watermark extraction process.

TABLE III

DESCRIPTIONS OF STANDARD TEST SEQUENCES

No	Test Seq.	Content Descrptions
1	es01	Vocal (Suzanne Vega)
2	es02	German speech
3	es03	English speech
4	sc01	Trumpet solo and orchestra
5	sc02	Orchestral piece
6	sc03	Contemporary pop music
7	si01	Harpsichord
8	si02	Castanets
9	si03	pitch pipe
10	sm01	Bagpipes
11	sm02	Glockenspiel
12	sm03	Plucked strings

TABLE IV

WATERMARK PAYLOAD FOR STANDARD TEST SEQUENCES

Test Seq.	es01	es02	es03	sc01	sc02	sc03
Payload (bits/s)	3	2	2	4	3	7
Test Seq.	si01	si02	si03	sm01	sm02	sm03
Payload (bits/s)	3	19	5	4	13	3

V. EXPERIMENT RESULTS

In this section, we present some experiment results demonstrating the AAZ-WM system performance. 12 standard test sequences for audio coding are used. Each of them is single-channel and sampled at 48kHz. The detailed description of each sequence is listed in Table III.

In the remaining content of this section, firstly, the watermark payload is tested, followed by experiments illustrating the robustness of the core layer and LLE layer watermark separately. Next, the scalability of watermark strength, one important feature of AAZ-WM is examined. We then turn to look at the impact of the watermark on audio fidelity and data size expansion. During all the experiments, the watermark embedding bandwidth (see Section IV-A) is set to 40, corresponding to frequency ranging from 0 Hz to 1800 Hz. The maximum number of bit-plane expansion θ_{expd} (see Section IV-A.3) is set to 4. Accordingly in the LLE layer the spreaded embedding bandwidth is 160 and each coefficient has one bit-plane expansion. The core layer bitrate is set to 128kbps, and the watermark global strength bound ε (see Section IV-A.1) is set to 1.

A. Watermark Payload

Watermark payload is measured as the maximum number of bits per second of watermark message embedded, without any extraction error in all three extraction scenarios for a 30-seconds sequence. The result for the 12 test sequences is shown in Table IV. From the results we observe that si02 and sm02 have relatively high payload compared to others because the feature of the two sequences greatly facilitates temporal masking effect. In contrast, the speech sequences es02 and es03 give lower payload.

B. Watermark Robustness

The watermark robustness is measured in terms of watermark message extraction BER. In this experiment, the robustness of the core layer watermark is demonstrated by i) MP3 and AAC compression of watermarked PCM audio (with LLE layer fully truncated), ii) downsampling (48kHz to 22kHz to 48kHz), iii) bandpass filtering (cut-off at 100Hz and 1300Hz), iv) echo addition and v) equalization. The robustness of the LLE layer watermark is demonstrated by transcoding of LLE layer bitstream. The results are shown in Table V. Note that in the test demo, we did not implement mechanism against attacks such as desynchronization, time and frequency fluctuation and *etc*. However, as addressed in the introduction part, as this proposed solution stands on the system level, many other SS-based watermarking algorithms can be directly applied to this system to improve performance.

TABLE V

WATERMARK MESSAGE EXTRACTION BER UNDER VARIOUS ATTACKS

Test Seq.	es01	es02	es03	sc01	sc02	sc03
No manipulation	0	0	0	0	0	0
MP3@128kbps	0	0	0	0	0	0
MP3@64kbps	0.029	0	0	0.032	0	0
AAC@128kbps	0	0	0	0	0	0
AAC@64kbps	0	0	0	0	0	0
Downsampling	0	0	0	0	0	0
Bandpass filtering	0	0.047	0	0	0	0
Echo addition	0	0.047	0	0.023	0	0
Equalization	0	0	0	0	0	0
LLE@128kbps	0	0	0	0	0	0
LLE@64kbps	0	0	0	0	0	0
LLE@32kbps	0	0	0	0	0	0
Test Seq.	si01	si02	si03	sm01	sm02	sm03
No manipulation	0	0	0	0	0	0
MP3@128kbps	0	0	0	0	0	0
MP3@64kbps	0.031	0	0.019	0	0	0
AAC@128kbps	0	0	0	0	0	0
AAC@64kbps	0	0	0	0	0	0
Downsampling	0.030	0	0	0	0	0
Bandpass filtering	0	0	0	0	0	0
Echo addition	0	0	0	0	0	0
Equalization	0	0	0	0	0	0
LLE@128kbps	0	0	0	0	0	0
LLE@64kbps	0	0.038	0	0	0	0
LLE@32kbps	0	0.090	0	0	0.028	0

(Note: LLE@128kbps refers transcoding attack such that the remaining LLE bitrate is 128kbps. The payload for each sequence has been set the same as in Table IV.)

C. Watermark Scalability

The watermark scalability is demonstrated in terms of the watermark strength as a function of the streaming bitrate (*i.e.*, core layer bitrate + LLE layer bitrate after transcoding), where the watermark strength is measured in terms of the average value of the extraction statistic magnitude $|\chi_1|$ over all watermark message bits. Note that this test applies to Extraction Scenario 1 only.

As demonstrated in Figure 9, the watermark strength decreases as the streaming bitrate increases, which matches what we have expected. This curve has similar characteristic as a general rate-distortion curve, since in general, the watermark strength is proportional to the distortion it introduces to the host signal.

D. Perceptual Quality

We have used PEAQ – the ITU standard for objective measurement of perceived audio quality – to test the watermark impact on the perceptual audio quality [30]. The output score is in terms of objective difference grade (ODG) in 0 to -4 scale, where 0 indicates the difference is imperceptible whereas -4

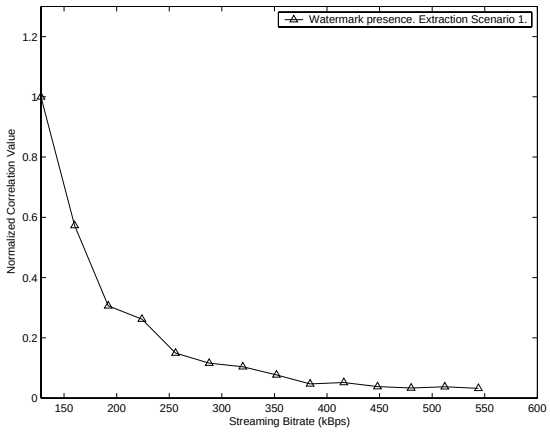

Fig. 9. Watermark strength against streaming bitrate for Extraction Scenario 1. The streaming bitrate is the sum of core layer bitrate (128kbps) and LLE layer bitrate.

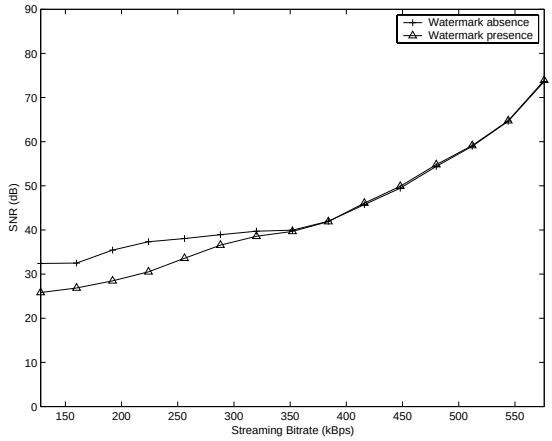

Fig. 10. Comparison of rate-distortion characteristics of i) watermark absence ii) watermark presence. The streaming bitrate is the sum of core layer bitrate (128kbps) and LLE layer bitrate.

TABLE VI
PERCEPTUAL AUDIO QUALITY MEASUREMENT BY PEAQ

Test Seq.	es01	es02	es03	sc01	sc02	sc03
ODG (AAZ core)	-0.57	-0.75	-1.03	-1.36	-0.76	-0.45
ODG (AAZ core+WM)	-0.60	-0.76	-1.03	-1.77	-0.81	-0.50
Test Seq.	si01	si02	si03	sm01	sm02	sm03
ODG (AAZ core)	-0.60	-1.38	-0.65	-0.2	-1.47	-0.50
ODG (AAZ core+WM)	-0.69	-1.99	-1.18	-0.57	-3.36	-0.53

indicates the audio is very annoying. Table VI shows the PEAQ test results. The AAZ watermarked audio is benchmarked by the AAZ core layer decoded audio.

The results show that in most of the cases, the AAZ-WM system produces watermarked audio which has similar perceived quality as the AAZ core layer audio. The only exceptions are sequence si02 and sm02. After repetitive listening of these two sequences, we have identified two possible causes: i) the HAS model in the current implementation is not accurate enough, leading to too optimistic watermark capacitance estimation (refer to Table IV). ii) It is noticed that in si02 and sm02, the data hiding ability heavily relies on temporal masking effect. In the current implementation version of AAZ, however, the short/long window switching function is disabled, resulting in prominent pre-echo effect. Therefore, the perceptibility problem here is irrelevant to the proposed system, but merely due to some implementation issues.

E. Rate-Distortion

The rate-distortion characteristic is now examined. Figure 10 presents the curves of decoding an unwatermarked AAZ bitstream and a watermarked AAZ-WM bitstream, respectively. The curves show that the watermarked audio has similar rate-distortion characteristic as the unwatermarked audio, except that at low bitrate, there is a degradation of about 5 dB. The SNR increases as the streaming bitrate increases, and eventually the audio becomes lossless when there is no transcoding taking place.

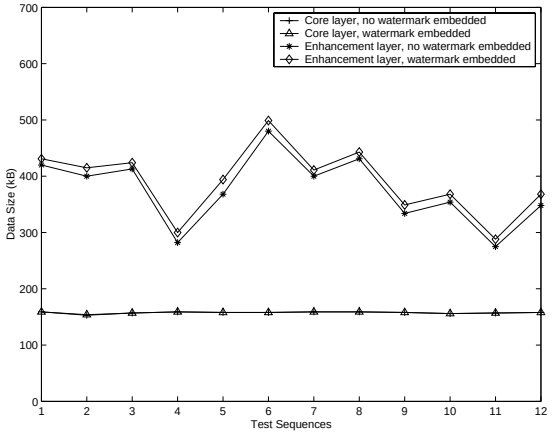

Fig. 11. Watermark impact on the data size. The transcoding rate is set to 0 such that the LLE bitstream is lossless.

F. Data Size Expansion

Figure 11 illustrates the data expansion properties. The design goal is to make the data expansion as small as possible. The results show that the watermark does not change the data size in the core layer, and only increase the data size in the LLE layer slightly. The bit-plane expansion (refer to Section IV-A.3) is a cause of increasing the data size. Note that the watermark embedding bandwidth (refer to Section IV-A) also determines how large the data size increases.

VI. CONCLUSIONS

In this paper, we present a novel approach to design lossless audio watermark based on a recently standardized two-layer scalable audio coder. The favorable features of this scheme - recovery of the original lossless audio, watermark adaptiveness and watermark scalability - are elaborated and experimentally demonstrated. Our main contributions include:

- Designed the lossless watermarking scheme in a scalable manner, i.e. incorporating the watermarking system in a layered scalable audio coder, such that the watermarking system inherits the scalability of the coder. In

this approach, we have ensured the perfect recovery of original lossless audio content after watermarking, and meanwhile, the watermark is made well adaptive to the content, leading to significant perceptual audio quality improvement. Compared with other lossless watermarking scheme which typically binds the watermark in those insignificant components, in this proposed method we are also able to make the watermark robust by embedding watermark into the significant component of the media content.

- Applied the ISS technique in a practical way to fine-tune the watermark distortion to improve the imperceptibility (Section IV-A.2). In addition, proposed to further reduce the watermark distortion by adaptively allocating watermark to the host signal coefficients (Section IV-A.3).
- Proposed a new method for detecting the presence of watermark in an audio content, in order to enhance the confidence level of watermark extraction (Section IV-B.2).

REFERENCES

[1] R.S. Yu, X. Lin, S. Rahardja, and C.C. Ko. A scalable lossy to lossless audio coder for mpeg-4 lossless audio coding. In *IEEE International Conference on Acoustics, Speechs, and Signal Processing*, 2004.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM Systems Journal*, 35, 1996.

[3] D. Gruhl and W. Bender. Echo hiding. In *Information Hiding Workshop*, Cambridge, U.K.

[4] L. Boney, A. H. Tewfik, and K. H. Hamdy. Digital watermarks for audio signals. In *EUSIPCO*, Trieste, Italy, September 1996.

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

[6] F. Hartung and B. Girod. Watermarking of uncompressed and compressed video. *Signal Processing, Special Issue on Copyright Protection and Access Control for Multimedia Services*, May 1998.

[7] M. Barni, F. Bartolini, V. Cappellini, and A. Piva. A dct-domain system for robust image watermarking. *Signal Processing, Special Issue on Copyright Protection and Access Control for Multimedia Services*, May 1998.

[8] M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney. Robust audio watermarking using perceptual coding. *Signal Processing, Special Issue on Copyright Protection and Access Control for Multimedia Services*, May 1998.

[9] I. J. Cox, M. L. Miller, and A. McKellips. Watermarking as communications with side information. *Proceedings of the IEEE*, July 1999.

[10] M. Arnold. Audio watermarking: Features, applications and algorithms. In *IEEE International Conference on Multimedia and Expo*, 2000.

[11] B. Chen and G. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47, May 2001.

[12] J. Fridrich, M. Goljan, and R. Du. Invertible authentication. In *SPIE Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, January 2001.

[13] M. van der Veen, F. Bruekers, J. Haitsma, T. Kalker, A.N.Lemma, and W. Oomen. Robust, multi-functional and high-quality audio watermarking technology. In *110th AES Convention*, Amsterdam, the Netherlands.

[14] N. Nikolaidis, V. Solachidis, A.Tefas, and I. Pitas. Watermark detection: Benchmarking perspectives. In *IEEE International Conference on Multimedia and Expo*, 2002.

[15] D. Kirovski and H. S. Malvar. Spread-spectrum watermarking of audio signals. *IEEE Transactions on Signal Processing*, Apr 2003.

[16] Z.M. Lu, W. Xing, D. G. Xu, and S. H. Sun. Digital image watermarking method based on vector quantization with labeled codewords. *IEICE Transactions on Information and Systems*, E86-D, Dec 2003.

[17] H. S. Malvar and D. A. F. Florencio. Improved spread spectrum: A new modulation technique for robust watermarking. *IEEE Transactions on Signal Processing*, 5(4), April 2003.

[18] Z. Liu and A. Inoue. Audio watermarking techniques using sinusoidal patterns based on pseudorandom sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, August 2003.

[19] I.-K.Yeo and H.-J.Kim. Modified patchwork algorithm: A novel audio watermarking scheme. *IEEE Transactions on Speech and Audio Processing*, July 2003.

[20] C. De Vleeschouwer, J. F. Delaigle, and B. Macq. Circular interpretation of bijective transformations in lossless watermarking for media asset management. *IEICE Transactions on Multimedia*, 5, March 2003.

[21] W. Trappe, M. Wu, Z.J. Wang, and K.J.R. Liu. Anti-collusion fingerprinting for multimedia. *IEEE Transactions on Signal Processing*, April 2003.

[22] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q.B. Sun, and X. Lin. Robust lossless image data hiding. In *IEEE International Conference on Multimedia and Expo*, 2004.

[23] R. Tachibana, S. Shimizu, S. Kobayashi, and T. Nakamura. Au audio watermarking method robust against time- and frequency-fluctuation. In *Security and Watermarking of Multimedia Contents III, SPIE*, San Jose, USA.

[24] Scalable lossless coding (sls). In *SC29/WG11/N6673*, Text of 14496-3:2001/PDAM 5, Seattle, USA, July, 2004.

[25] M. Bosi and et al. Iso/iec mpeg-2 advanced audio coding. In *J. Audio Eng. Soc.*, Vol. 45, No.10, pp. 789-814, 1997.

[26] R.S. Yu, C.C. Ko, S. Rahardja, and X. Lin. Bit-plane golomb code for sources with laplacian distributions. In *IEEE International Conference on Acoustics, Speechs, and Signal Processing*, 2003.

[27] S. Wee and J. Apostolopoulos. Secure transcoding with jpsec confidentiality and authentication. In *IEEE International Conference on Image Processing*, 2004.

[28] I.J.Cox, M.L.Miller, and J.A.Bloom. *Digital Watermarking*. Morgan Kaufman Publishers, 2001.

[29] R. Sugihara. Practical capacity of digital watermark as constrained by reliability. In *IEEE International Conference on Information Technology: Coding and Computing*, Las Vegas, USA, 2001.

[30] T. Thiede *et al.* Peaq – the itu standard for objective measurement of perceived audio quality. *Journal of the Audio Engineering Society*, January/Feburary 2000.

APPENDIX
LAYER I WATERMARK SIGNAL AT LOSSY STREAMING RATE

In this section, we derive Equation 16. Assume $i_w(k)$ and $e_w(k)$ can be represented in L_i and L_e bits, respectively. The bitstream sampling rate is R_0. Therefore, the Layer I and Layer II bitrates before entropy coding are $L_i R_0$ and $L_e R_0$, respectively. Denote the transcoding rate and normalized transcoding rate by R_{trc} and $\hat{R}_{trc}$. We have:

$$0 \leq R_{trc} \leq L_e R_0 \tag{38}$$

or:

$$0 \leq \hat{R}_{trc} \leq L_e \tag{39}$$

where $\hat{R}_{trc} = R_{trc}/R_0$. At transcoding rate $\hat{R}_{trc}$, the truncated residue is:

$$e_w(k, \hat{R}_{trc}) = int[e_w(k)/2^{\hat{R}_{trc}}]2^{\hat{R}_{trc}} \tag{40}$$

From Equation 11, the reconstructed coefficient is:

$$c(k, \hat{R}_{trc}) = int[e_w(k)/2^{\hat{R}_{trc}}]2^{\hat{R}_{trc}} + Q^{-1}[i_w(k)] \tag{41}$$

$c(k, \hat{R}_{trc})$ is to be requantized before watermark extraction. We have:

$$\begin{aligned} & i_w(k) + n_e(k, \hat{R}_{trc}) = Q[c(k, \hat{R}_{trc})] \\ & = Q\{int[e_w(k)/2^{\hat{R}_{trc}}]2^{\hat{R}_{trc}} + Q^{-1}[i_w(k)]\} \\ & = Q\{int[e_w(k)/2^{\hat{R}_{trc}}]2^{\hat{R}_{trc}}\} + i_w(k) \\ & = int\{int[e_w(k)/2^{\hat{R}_{trc}}]2^{\hat{R}_{trc}}/\lambda\} + i(k) + B\beta_c b(k) \end{aligned} \tag{42}$$

From Equation 21,

$$\begin{aligned} & i_w(k) + n_e(k, \hat{R}_{trc}) \\ &= int\{int[(e(k) - \lambda B\beta_c b(k))/2^{\hat{R}_{trc}}]2^{\hat{R}_{trc}}/\lambda\} \\ &\qquad + i(k) + B\beta_c b(k) \\ &= int\{int[-\lambda B\beta_c b(k)/2^{\hat{R}_{trc}}]2^{\hat{R}_{trc}}/\lambda\} + i(k) + B\beta_c b(k) \\ &= -int[\lambda B\beta_c b(k)/2^{\hat{R}_{trc}}]2^{\hat{R}_{trc}}/\lambda + i(k) + B\beta_c b(k) \\ &= i(k) + \{B\beta_c b(k) - int[B\beta_c b(k)/(2^{\hat{R}_{trc}}/\lambda)](2^{\hat{R}_{trc}}/\lambda)\} \\ &= i(k) + \psi(k, \hat{R}_{trc}) \leq i(k) + 2^{\hat{R}_{trc}}/\lambda \end{aligned} \tag{43}$$

where $\psi(k, \hat{R}_{trc})$ can be seen as the reminder of $B\beta_c b(k)$ divided by $2^{\hat{R}_{trc}}/\lambda$. When $\hat{R}_{trc} = L_e$, Layer II is fully truncated. In this case, $\psi(k, \hat{R}_{trc}) = B\beta_c b(k) \leq 2^{\hat{R}_{trc}}/\lambda$. If we assume equality, we have:

$$\begin{aligned} i_w(k) + n_e(k, \hat{R}_{trc}) &= i(k) + 2^{\hat{R}_{trc}}/\lambda \\ &= i(k) + (2^{L_e}/\lambda)2^{\hat{R}_{trc} - L_e} \\ &= i(k) + B\beta_c b(k)2^{\hat{R}_{trc} - L_e} \end{aligned} \tag{44}$$

LAYER II (LLE) WATERMARK EXTRACTION CRITERIA FOR NON-UNIFORM QUANTIZER

In this section, the LLE layer watermark extraction criteria described in Equation 9 and 10 are derived for the non-uniform quantizer in Equation 1 and 2. For simplicity, let us replace 0.4054 by 0. The property of the quantizer will remain similar. After de-multiplexing and BPGC decoding, we obtain the watermarked residue $\mathbf{e}'$. Similar to the derivation of Equation 21, now we have:

$$\begin{aligned} \Delta(k) &= e_w(k) - e(k) \\ &= [c(k) - Q^{-1}(i_w(k))] - [c(k) - Q^{-1}(i(k))] \\ &= \frac{1}{\alpha}(\sqrt[4]{2^{SF(c(k))}}|i(k)|^{4/3})sgn[i(k)] \\ &\quad - \frac{1}{\alpha}(\sqrt[4]{2^{SF(c(k))}}|i_w(k)|^{4/3})sgn[i_w(k)] \end{aligned} \tag{45}$$

Equation 28, 29 and 31 shows $sgn[i_w(k)] = sgn[i(k)]$. Therefore,

$$\begin{aligned} \Delta(k) &= \frac{1}{\alpha}(\sqrt[4]{2^{SF(c(k))}})[(i(k))^{4/3} - (i_w(k))^{4/3}]sgn[i(k)] \\ &= \frac{1}{\alpha}(\sqrt[4]{2^{SF(c(k))}})* \\ &\quad [(i(k))^{4/3} - (i(k) + b(k)\beta(k))^{4/3}]sgn[i(k)] \end{aligned} \tag{46}$$

Using binomial expansion, we have:

$$[i(k) + b(k)\beta(k)]^{4/3} \approx i(k)^{4/3} + \frac{4}{3}b(k)\beta(k)i(k)^{1/3} \tag{47}$$

for $|\frac{i(k)}{b(k)\beta(k)}| > 1$.Note that $i(k)^{1/3}sgn[i(k)] = |i(k)|^{1/3}$. We have,

$$\Delta(k) \approx -\frac{4}{3\alpha}\sqrt[4]{2^{SF(c(k))}}\beta(k)|i(k)|^{1/3}b(k) \tag{48}$$

$e_w(k)$ can thus be expressed as:

$$\begin{aligned} e_w(k) &= e(k) + \Delta(k) \\ &= e(k) + [-\frac{4}{3\alpha}\sqrt[4]{2^{SF(c(k))}}\beta(k)|i(k)|^{1/3}]b(k) \\ &= e(k) + \phi(k)b(k) \end{aligned} \tag{49}$$

Both $e(k)$ and $\phi(k)$ are independent of $b(k)$. In addition, $\phi(k)$ is always negative. Hence, we can use Equation 9 and 10 to extract watermark bits for the non-uniform quantizer.