A Scalable Watermarking Scheme for the Scalable Audio Coder

[1, 2]Z. Li, [1]Q.B. Sun, [2]Y. Lian and [1]R.S. Yu

[1] Institute for Infocomm Research (I^2R)
21 Heng Mui Keng Terrace, Singapore 119613
[2] Department of Electrical and Computer Engineering
National University of Singapore
10 Kent Ridge Crescent, Singapore 119260

***Abstract*—In this paper, we describe a scalable (i.e., lossless-to -lossy) watermarking scheme which overcomes the problem of non-invertible distortion introduced by the watermark signal. The scheme is based on a standardized scalable audio coder [9] – as a result, the embedded watermark inherits the scalability of the audio coder. We elaborate how the scalability can be used to realize the recovery of the lossless audio signal after watermarking. The experimental results demonstrate the validity of the proposed watermarking scheme in terms of robustness, data expansion and perceptual quality.**

***Keywords*—Watermark, scalable audio coding, spread spectrum**

I. Introduction

Early development of digital watermarking includes schemes such as hiding data in the least significant bits (LSB) of the host signal. Such schemes have been proven to be too susceptible to noise and attacks. Later, more advanced schemes have been proposed in literature. To our knowledge, various advanced watermarking schemes can be roughly classified into two categories: spread spectrum (SS) based watermarking schemes [1][2][3] and quantization based watermarking schemes [4][5]. SS-based watermarking schemes inherit the robustness against AWGN and other attacks from traditional SS communications. However, it usually fails to reject the influence of the host signal, resulting limited data payload. Quantization based watermarking schemes immune from the influence of the host signal, and thus has large data payload. However, it suffers from attacks like magnitude scaling. In addition, codebook design has been a difficult job up to date.

One common drawback the basic watermarking schemes suffered from (including both SS-based and quantization based) is: the distortion introduced by the watermark signal is non-invertible. That is, once the watermark is embedded, one is unable to recover the original lossless signal. We call them *lossy* watermarking schemes. This motivates to develop *lossless* watermarking schemes in which the original signal is still recoverable even after the watermark embedding. To solve this problem, one approach called *lossless / invertible* watermarking has been proposed [6][7][8]. In their approaches, the exact original signal can be recovered after watermark extraction. However, the embedded watermark is not bound well with the original signal which may affect the watermark robustness. In this paper, we shall discuss an alternative approach – *scalable* watermarking which aims to bridge lossless watermarking and lossy watermarking. We elaborate how the original lossless signal can be recovered by using the scalable watermarking scheme.

The proposed framework is based on the Advanced Audio Zip (AAZ) coder [9], which has been included in the Commission Draft for on-going scalable audio coding standard under MPEG4 [10]. We therefore call our system AAZ-WM. The AAZ coder has two layers – the core layer and the lossless enhancement layer. The core layer is essentially an embedded Advanced Audio Coding (AAC) [11] encoder (backward compatibility) serving as the basic bitstream whereas the enhancement layer compensates the quantization loss of the core layer. The enhancement layer data are further Bit-Plane-Golomb-Coded (BPGC). By truncating the enhancement layer data at arbitrary bit-plane, different streaming bitrates from perceptually lossy to lossless can be met. Our AAZ-WM is embedded in the AAZ coder, with the ability of watermarking both layers. The watermarks in the two layers are designed in such a way that the watermark in the enhancement layer compensates the watermark in the core layer. That is, when the enhancement layer is not truncated at all, the original lossless signal can be recovered. Scalability of watermarking can be achieved with different truncation (i.e., transcoding) rates in the enhancement layer: when the enhancement layer is fully truncated, the watermark strength is the strongest; when the enhancement layer fully compensates the core layer, the watermark is fully compensated and thus the original lossless signal can be recovered.

Besides the lossless-recovery feature, the scalability of AAZ-WM can be used in other ways. Potential applications based on AAZ-WM are suggested as follows: a) Copyrights protection applications for online audio streaming. The watermarking scalability could flexibly protect the audio based on the received quality (i.e., Quality of Protection [12]) (Refer to Section III-D: Watermark Decoding Scenario 1),2),3)). b) Broadcast monitoring (Refer to Decoding Scenario 2)). AAZ-WM can be used for detecting the air time of advertisement broadcasted by online radio stations. c) Blind audio quality authentication (Refer to Decoding Scenario 1)). Since the watermark strength changes as the enhancement layer bitrate changes, the watermark strength can be used to authenticate the quality of audio without the presence of the original audio file. d) Lyrics captioning (Refer to Decoding

Scenario 2)). When streaming music online, it is desirable to have the lyrics transmitted simultaneously with the audio stream. Using AAZ-WM, the lyrics can be encoded in the watermark and transmitted with the AAZ bitstream.

In Section II, we firstly introduce a generic model of our watermarking system for layered scalable coders. In Section III, we describe the complete AAZ-WM algorithm and address some practical issues regarding the system implementation in the AAZ coder. In Section IV, we present some experiment results. Conclusions are drawn in Section V.

II. A GENERIC SYSTEM MODEL

A. *Notations*

Firstly let us introduce some notations used in this paper:

Vectors are denoted $\bar{a}$. Elements of the vector are denoted a_k . As an example, $\bar{a} = \{a_1, a_2, a_3 ...\}$.

Quantized quantities are denoted $\tilde{a}$.

Watermarked quantities are denoted a' .

Quantities with unknown property is denoted $\breve{a}$.

B. *The Generic System Model*

The generic watermark embedding system for layered scalable coder is shown in Fig. 1.

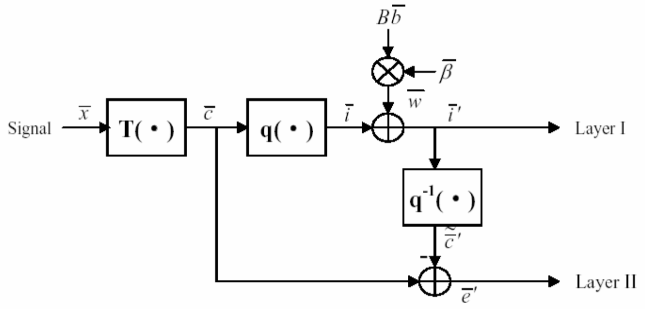

Fig. 1. Generic Watermark Embedding System for Layered Scalable Coder

In the layered scalable coder, the signal $\bar{x}$ is firstly transformed into frequency domain coefficients $\bar{c}$ via **T(•)** operation. For achieving coding gain, the coefficients $\bar{c}$ is then quantized by **q(•)** before being entropy coded in Layer I. Therefore,

$$\bar{i} = q(\bar{c}) = q[T(\bar{x})] \tag{1}$$

In Layer II, the residue $\bar{e}$ is found by

$$\bar{e} = \bar{c} - q^{-1}\left[\bar{i}\right] = \bar{c} - q^{-1}\left[q(\bar{c})\right] \tag{2}$$

Where **q^{-1}(•)** is the inverse quantizer. The residue $\bar{e}$ is used to compensate the quantization distortion in the decoder.

For our watermark embedding system, the watermark is embedded to the quantized coefficients in Layer I. Note that the watermark must be *integer* values. The message bit *B* is firstly spread by a spreading sequence $\bar{b}$, which has *K* chips and is generated from a secret key. The spread signal is then perceptually shaped by the local watermark strength, $\bar{\beta}$, where $\beta_k \geq 0$ and β_k is an integer. Therefore the watermark is $\bar{w} = B\bar{b} \circ \bar{\beta}$, where $\bar{a} \circ \bar{b} = \{a_1 b_1, a_2 b_2, ...\}$ denotes array multiplication.

The watermarked quantized coefficients are:

$$\bar{i}' = \bar{i} + \bar{w} = q(\bar{c}) + B\bar{b} \circ \bar{\beta} \tag{3}$$

Now the residue $\bar{e}'$ is then given by:

$$\bar{e}' = \bar{c} - q^{-1}\left[\bar{i}'\right] = \bar{c} - q^{-1}\left[q(\bar{c}) + B\bar{b} \circ \bar{\beta}\right] \tag{4}$$

Note that $\bar{e}'$ is also watermarked.

The decoding of message bit in Layer I is done by correlating the received coefficients and the spreading sequence:

$$\chi_1 = < \bar{\breve{i}}, \bar{b} > \tag{5}$$

Where $< a, b > = (1/K)\sum_{k=1}^{K} a_k b_k$ is the normalized inner product. The estimated bit is:

$$\hat{b}_1 = \operatorname{sgn}(\chi_1) \tag{6}$$

Similarly, for Layer II, the detection criteria is

$$\chi_2 = < \bar{\breve{e}}, \bar{b} > \tag{7}$$

$$\hat{b}_2 = -\operatorname{sgn}(\chi_2) \tag{8}$$

In order to recover the original lossless signal, we need to perfectly reconstruct the frequency domain coefficients $\bar{c}$. In the decoder, this is simply done by:

$$\bar{c} = \bar{e}' + q^{-1}\left[\bar{i}\right] \tag{9}$$

III. DESCRIPTION OF THE AAZ-WM SYSTEM

In the previous section, we have built up a generic watermarking model for layered scalable audio coder. In this section, we introduce the complete AAZ-WM system. Section III-A describe the structure of the AAZ-WM encoder / embedder. Section III-B and III-C jointly give an upper bound for the watermark strength in order to meet the requirement of fidelity and lossless recovery. The watermark message decoding is finally described in Section III-D.

A. *AAZ-WM Encoder / Embedder*

The complete structure of the AAZ-WM is shown in Fig. 2.

Refer to Fig.2, in the original AAZ encoder, the original audio signal is firstly transformed into frequency domain coefficients using integer MDCT (intMDCT). Let us denote the frequency domain coefficients $\bar{c}$, where 1024 elements c_k form one intMDCT block. 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. The scale-factor of the band where c_k belongs to is denoted $SF(c_k)$. $\bar{c}$ is further normalized by a isotropic factor α in order to approximate the outputs of the MDCT filterbank used in AAC [9].

A non-uniform quantizer **q(•)** is used in AAZ [9][10]:

$$q(c_k) = i_k = \left\lfloor \left(\frac{|\alpha c_k|}{2^{SF(c_k)/4}}\right)^{3/4} + 0.4054 \right\rfloor \operatorname{sgn}(c_k) \qquad (10)$$

Where $\lfloor \bullet \rfloor$ denotes flooring operation and $\operatorname{sgn}(\bullet)$ denotes sign operation. $\bar{i}$ is the quantized intMDCT (QintMDCT) coefficients . Similarly, $\mathbf{q}^{-1}(\bullet)$ is:

$$q^{-1}(i_k) = \tilde{c}_k = (1/\alpha)\left(2^{SF(C_k)/4}(|i_k| - 0.4054)^{4/3}\right)\operatorname{sgn}(i_k) \qquad (11)$$

Note that in Section II, we introduced the detection criteria as in (5)(6) and (7)(8). It is not difficult to prove these detection criteria are applicable if $\mathbf{q}(\bullet)$ and $\mathbf{q}^{-1}(\bullet)$ are linear quantizers. The non-uniform quantizer used in AAZ actually has very similar property as a linear quantizer, and thus the same detection criteria can be applied too. We shall provide both mathematical analysis and experiment results to illustrate our arguments in the remaining content.

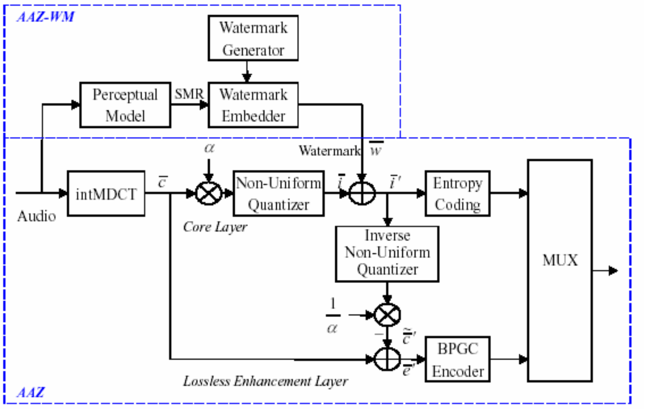

Fig. 2. Structure of the AAZ-WM Encoder/Embedder

In AAZ-WM, the watermark $\bar{w}$, generated from the watermark embedder module, is added to $\bar{i}$. The watermark is chosen to be added to the perceptually significant band (i.e. the near-DC components) of the spectrum so that it cannot be removed by attackers without sacrificing the fidelity. In our experiment, the watermark is added to the first 20 QintMDCT coefficients, corresponding to frequency ranging from 0 Hz to 900 Hz.

In Layer I (the core layer), the watermarked QintMDCT coefficients are entropy-coded to form the bitstream. In Layer II (enhancement layer), the watermarked residues are BPGCed. At the last stage, the two bitstreams in the core layer and the enhancement layer, together with the other side information such as the scale-factors are multiplexed and form the final bitstream.

B. *Computing Distortion Bounds from the Perceptual Model*

In order to enhance the audio fidelity, a perceptual model is used to find a bound of inaudible distortions by human ears. The allowable distortion is computed in terms of signal-to-masking ratio (SMR) for each intMDCT coefficient. The SMR for coefficient c_k is denoted by $SMR(c_k)$.

Firstly, the quantized intMDCT coefficients $\tilde{c}_k$ are reconstructed using (11). For a scale factor band (corresponding to coefficient indices from k_1 to k_2), the total energy is $\sum_{k=k_1}^{k_2} c_k^{\ 2}$, therefore the total allowable distortion is $\left(\sum_{k=k_1}^{k_2} c_k^{\ 2}\right) / SMR(c_k)$. For the allowable distortion for each coefficient, it is desirable to make it proportional to the coefficient, i.e. the distortion is δc_k , where δ is a constant. Therefore, the total allowable distortion within a scale-factor band is:

$$\sum_{k=k_1}^{k_2} (\delta c_k)^2 = \left(\sum_{k=k_1}^{k_2} c_k^{\ 2}\right) / SMR(c_k)$$

$$\delta = 1/\sqrt{SMR(c_k)} \; . \qquad (12)$$

Hence, the distortion bounds are:

$$c_{-,k} = \tilde{c}_k (1 - \varepsilon / \sqrt{SMR(c_k)})$$

$$c_{+,k} = \tilde{c}_k (1 + \varepsilon / \sqrt{SMR(c_k)}) \qquad (13)$$

where ε is a global strength bound. When ε =1, the allowable distortion is controlled within one unit of just noticeable distortion (JND). For low bitrate where the major distortion is due to quantization error, ε is set to be larger than 1. The distortion bounds are then converted to the bounds for i_k :

$$i_{-,k} = q(c_{-,k}) \; , \; i_{+,k} = q(c_{+,k}) \qquad (14)$$

C. *Problem Related to Computing Max_Bitplane for BPGC*

In coding the residue using BPGC, one important step is to compute the number of bit-planes needed for the residue e_k . In the original AAZ coder [9][10], residue value is bounded by:

$$0 \le e_k < residue\_bound_k \qquad (15)$$

where

$$residue\_bound_k = (2^{SF(C_k)/4} / \alpha)\left[(|i_k| + 1 - 0.4054)^{4/3} - (|i_k| - 0.4054)^{4/3} \right]$$

Hence, the maximum number of bit-planes is bounded by:

$$max\_bitplane_k \le \log_2 (residue\_bound_k) \qquad (16)$$

In order to decode the original lossless audio file, it is necessary to compute the maximum number of bit-planes correctly. However, when the watermark is added, the residue e'_k (instead of e_k) is no longer bounded by *residue_bound* as in (15), i.e., the *max_bitplane* is not bounded.

As a solution, we define a global constant BP_EXPAND _NUMBER, which is used to constrain the maximum number of bit-planes. For example, if we set BP_EXPAND_NUMBER to 3, it suggests approximately three more bit-planes are used to encode the residues, assuming a linear quantizer is employed. But since the quantizer is non-uniform, the actually number of bit-plane needed may vary slightly throughout the coding.

Knowing BP_EXPAND_NUMBER in advance, the *residue_ bound* can be calculated in the encoder and decoder as follows:

$$residue\_bound_k = (2^{SF(C_k)/4}/\alpha)[(|i_k| + 2^{BP\_EXPAND\_NUMBER} - 0.4054)^{4/3} - (|i_k| - 0.4054)^{4/3}] \tag{17}$$

This is equivalent to allowing the added watermark to take the integer value between $\pm\,(2^{BP\_EXPAND\_NUMBER} - 1)$. E.g. for BP_EXPAND_NUMBER=3, the watermark can take values from -7 to +7 while the lossless audio can still be recovered. If the watermark value exceeds ± 7, we truncate the value to ± 7. Therefore, BP_EXPAND_NUMBER also serves as a bound controlling the watermark strength.

To summarize, the watermark strength is bounded both by the perceptual model shaping and BP_EXPAND_NUMBER. Therefore, the overall bound for i_k' is:

$$\begin{aligned}&\max(i_k - 2^{BP\_EXPAND\_NUMBER} + 1, i_{-,k}) \\ &\quad \le i_k' \le \min(i_k + 2^{BP\_EXPAND\_NUMBER} - 1, i_{+,k})\end{aligned} \tag{18}$$

Knowing this bound, we can control the perceptual distortion by shaping the watermark strength prior to adding it to the host signal.

D. Watermark Message Decoding

The decoding of the watermark message bits could be performed in three scenarios, including: *1)* Message decoding after audio decoding. *2)* Message decoding in the core layer before audio decoding. *3)* Message decoding in the enhancement layer before audio decoding.

1) Message decoding after audio decoding

In the decoder, the bitstream in the core layer is decoded and then compensated by the decoded residue in the enhancement layer to form the final audio stream. The residue can be truncated at arbitrary bit-plane to obtain the wanted bitrate. When the residue is not truncated at all, the original lossless audio can be recovered. We can expect in this case, the watermark is totally "removed". Therefore, the watermark power (and thus the decoding bit error rate) depends on the truncation rate of the encoded audio.

Fig. 3 illustrates the structure of the decoder. $\breve{\bar{c}}$ is the intMDCT coefficients of the unknown received audio file. $\breve{\bar{c}}$ is quantized using (10).

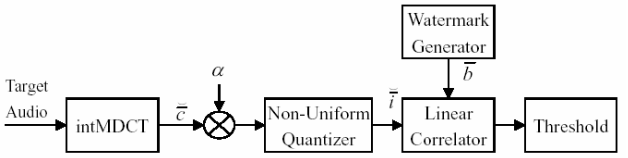

Fig. 3. Watermark Decoding after Audio Decoding

Note that the scale-factor for c_k in (10) is not available to the watermark decoder for blind decoding (i.e. the original signal is not available at the decoder). However, since the received audio and the original audio should be perceptually transparent, they should have similar characteristics in the *quantization and coding* process, hence, having similar scale-factor values. Therefore, the received audio is used to calculate the scale-factor to approximate the original file. It has been tested by experiments that this approximation is good enough for watermark message decoding.

The remaining decoding process are described in (5) and (6).

2) Message decoding in the core layer before audio decoding

Watermark decoding in the core layer is very similar to D.1), except now we recover the QintMDCT coefficients from the AAZ bitstream (refer to Fig. 4). The AAZ bitstream is firstly de-multiplexed. The bitstream of the core layer is entropy -decoded to recover $\bar{i}'$. The remaining process is similar to what is done in D.1).

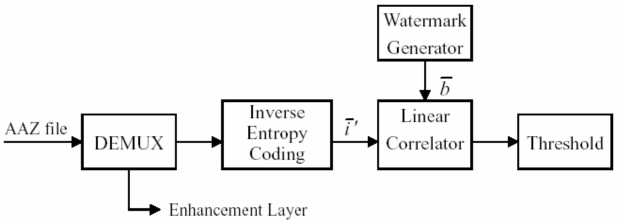

Fig. 4. Watermark Decoding in the core layer Before Audio Decoding

3) Message decoding in the enhancement layer before audio decoding

We now derive the watermark decoding criteria in the enhancement layer for the non-uniform quantizer in (10) and (11). For simplicity, let us replace 0.4054 by 0. The property of the quantizer will remain the same. After de-multiplexing and BPGC decoding, the bitstream we obtain is $\bar{e}'$.

Let $\bar{\Delta}$ denote the difference between the watermarked and unwatermarked residue. We have:

$$\begin{aligned}\Delta_k &= e_k' - e_k = (c_k - \tilde{c}_k') - (c_k - \tilde{c}_k) \\ &= \tilde{c}_k - \tilde{c}_k' = (1/\alpha)\left(2^{SF(C_k)/4}|i_k|^{4/3}\right)\mathrm{sgn}(i_k) \\ &-(1/\alpha)\left(2^{SF(C_k)/4}|i_k'|^{4/3}\right)\mathrm{sgn}(i_k')\end{aligned}$$

Eq. (13)(14)(18) shows $\mathrm{sgn}(i_k') = \mathrm{sgn}(i_k)$. Therefore,

$$\begin{aligned}\Delta_k &= (1/\alpha)2^{SF(C_k)/4}\left[(i_k)^{4/3} - (i_k')^{4/3}\right]\mathrm{sgn}(i_k) \\ &= (1/\alpha)2^{SF(C_k)/4}\left[(i_k)^{4/3} - (i_k + b_k\beta_k)^{4/3}\right]\mathrm{sgn}(i_k)\end{aligned}$$

Using binomial expansion, we have

$$(i_k + b_k\beta_k)^{4/3} \cong (i_k)^{4/3} + (4/3)(b_k\beta_k)(i_k)^{1/3},$$

for $|i_k/(b_k\beta_k)| > 1$.

Therefore,

$$\Delta_k \cong (1/\alpha)2^{SF(C_k)/4}\left[-(4/3)(b_k\beta_k)\right]|i_k|^{1/3}$$

e_k' can thus be expressed as:

$$\begin{aligned}e_k' &= e_k + \Delta_k = e_k + (1/\alpha)2^{SF(C_k)/4} * \\ &\left[-(4/3)|i_k|^{1/3}\beta_k\right]b_k = e_k + \phi_k b_k\end{aligned} \tag{19}$$

Both e_k and ϕ_k are independent of b_k. In addition, ϕ_k is always negative. Hence, we can still use (7) and (8) to decode watermark bits for non-uniform quantizers. The block diagram of the decoder is shown in Fig. 5.

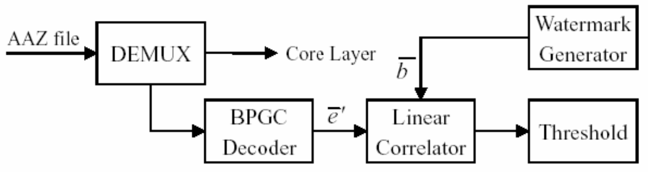

Fig. 5. Watermark Decoding in the Enhancement Layer Before Audio Decoding

IV. EXPERIMENT RESULTS

We have conducted a set of experiments to evaluate the performance of the proposed AAZ-WM. Since *robustness* and *payload* are two important measurements of any watermarking schemes, we firstly focus on testing the decoding bit error rate (BER) under different watermark payload (i.e. bitrate of watermark message). Next, the scalability of watermark strength, one important feature of AAZ-WM is examined. We then turn to test the impact of the watermark on the audio quality and file size.

A. Bit Error Rate in Three Decoding Scenarios

BER is tested for different watermark payload for three decoding scenarios (Fig. 6). For example, for a piece of audio of 30s length, the total embedded watermark message is 30 bits, assuming the watermark payload is 1 bit/sec. Since we only focus on testing the relationship between BER and watermark payload, for the scenario of watermark decoding after audio decoding, we fully truncate the enhancement layer to make the watermark power maximum.

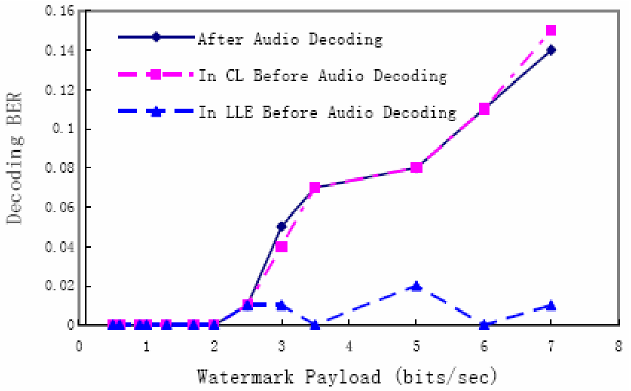

Fig. 6. Bit Error Rate vs. Payload in Three Decoding Scenarios. The target file is decoded with core layer bitrate = 128kbps and enhancement layer bitrate = 0kbps (i.e. maximum watermark power)

When the enhancement layer is not fully truncated, we can expect the BER is higher than this (further experiment on the case when enhancement layer is not fully truncated is performed in Section IV-C.). It is observed that when the payload is under 2 bits/sec, the decoded watermark message is error-free. In addition, the BER for decoding in the enhancement layer before audio decoding is much smaller than the other two cases. This is because the power of the residue signal is much smaller than that of the host signal. Therefore, in some scenarios it is advantageous to decode watermark bits in the enhancement layer instead of the core layer.

B. Bit Error Rate under Attacks (for Decoding Scenario 1 only)

The BER against watermark payload is then tested under attacks of MP3 compression (128kbps) and resampling (44.1kHz -> 20kHz -> 44.1kHz) (Fig. 7). The results show the BER under attack have similar curves as unattacked case, thus this system is robust against such attacks.

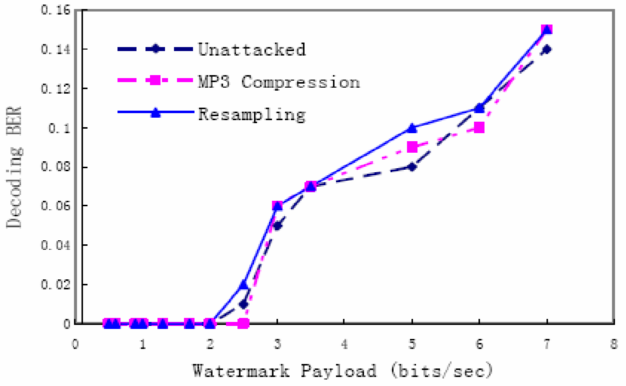

Fig. 7. Bit Error Rate vs. Payload Under Attacks. The target file is decoded with core layer bitrate = 128kbps and enhancement layer bitrate = 0kbps (i.e. maximum watermark power)

C. Testing the Scalability of Watermark Strength (for Decoding Scenario 1 only)

To test the scalability of watermark strength as the streaming bitrate varies (i.e., transcoding), we concatenate the spreading sequences for all bits and find the linear correlation over all bits duration. (Fig. 8) This correlation value serves as a better statistic which measures the watermark strength than BER, for Application c) mentioned in Section I. The tested audio is 5 second in length. Refer to Fig. 8, the point at 128kbps of streaming bitrate corresponds to the scenario of watermark decoding after audio decoding in Section IV-A. The result shows the watermark strength decreases as the bitrate in the enhancement layer increases, which agrees with what we have expected.

D. Signal Quality Test: SNR vs. Streaming Bitrate

Now we conduct the SNR test, which serves as an objective measurement of audio signal quality. The SNR is measured by comparing the compressed and watermarked signal against the original signal. Fig. 9 shows the watermarked audio signal have very similar audio quality as normal AAZ- compressed audio, except at low bitrate, there is a degradation of about 5 dB. However, since we have used the perceptual model as the audio quality measurement in our system, the 5dB degradation should be merely due to the changes in perceptually insignificant components, and thus does not harm the audio quality perceived by human ears.

E. Testing Watermark Influence on the File Size

We then test the impact of watermark on the file size for 9 pieces of audio files, each of length 5s. The core layter bitrate is fixed at 128kbps; the enhancement layer is without any truncation. The results in Table I show that the watermark does not change the data size in the core layer, and only increase the data size in the enhancement layer slightly.

V. CONCLUSIONS

In this paper, we presented a novel scalable watermarking (lossless-to-lossy) scheme. It embeds watermarks into a scalable audio coder, and the direct benefit is that the watermarking scheme inherits the scalability of the audio coder. We have illustrated how this scalability can be utilized to design the scalable watermarking scheme. Moreover, the watermarking system is design in such a way that the perceptual distortion is carefully controlled. Further work can be conducted regarding improving the system robustness and data payload.

REFERENCES

[1] L. Boney, A. H. Tewfik, and K. H. Hamdy, “Digital watermarks for audio signals,” in *Proc. EUSIPCO 1996*, Trieste, Italy, Sept. 1996.

[2] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, “Secure Spread Spectrum Watermarking for Multimedia,” *IEEE Trans. Image Processing*, Vol. 6, No.12, pp. 1673-1687, 1997.

[3] H. S. Malvar, D. A. F. Florencio, “Improved Spread Spectrum: A New Modulation Technique for Robust Watermarking”, *IEEE Trans. Signal Processing*, Vol. 5, No. 4, April 2003.

[4] B. Chen, G. Wornell, “Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding”, *IEEE Trans. Inform. Theory*, vol. 47, pp. 1423-1443, May 2001.

[5] 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*, Dec. 2003, Vol.E86-D, No.12, Paper ID: 2002EDL8089.

[6] J. Fridrich, M. Goljan and R. Du, “Invertible Authentication”, in *Proc. SPIE Security and Watermarking of Multimedia Contents*, San Jose CA, Jan. 23-26, 2001.

[7] J. Domingo-Ferrer, F Sebe, “Invertible Spread-spectrum Watermarking for Image Authentication and Multilevel Access to Precision-critical Watermarked Images”, *Proc. IEEE Intl. Conf. on Information Technology: Coding and Computing*, 2002.

[8] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q.B. Sun and X. Lin, “Robust Lossless Image Data Hiding”, *Proc.IEEE Intl. Conf. on Multimedia and EXPO*, 2004.

[9] R.S. Yu, X. Lin, S. Rahardja and C.C. Ko, “A Scalable Lossy to Lossless Audio Coder for MPEG-4 Lossless Audio Coding”, *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2004

[10] “Scalable Lossless Coding (SLS)”, *SC29/WG11/N6673, Text of 14496-3:2001/PDAM 5*, Seattle, USA, July, 2004

[11] M. Bosi and et al, “ISO/IEC MPEG-2 Advanced Audio Coding”, J. Audio Eng. Soc., Vol. 45, No.10, pp. 789-814, 1997 OCT.

[12] C. S. Ong, K. Nahrstedt and W. Yuan, “Quality of Protection for Mobile Multimedia Applications”, *Proc.IEEE Intl. Conf. on Multimedia and EXPO* 2003.

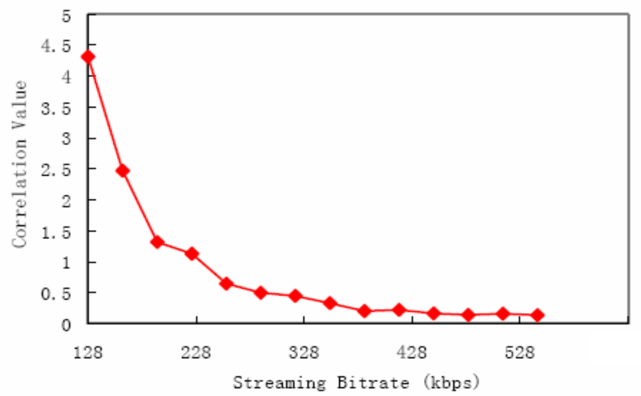

Fig. 8. Correlation Value vs. Streaming Bitrate. The streaming bitrate is the sum of core layer bitrate (fixed at 128kbps) and enhancement layer bitrate (from 0kbps onwards)

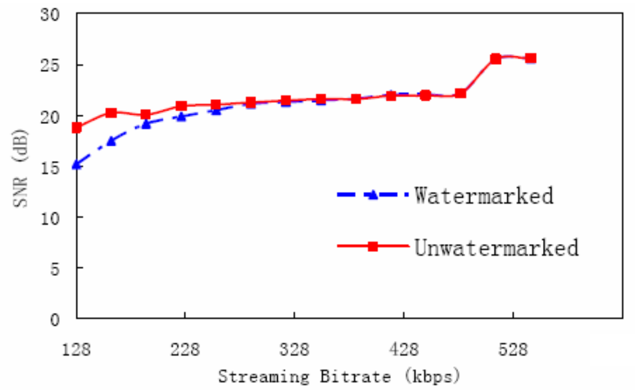

Fig. 9. SNR vs. Streaming Bitrate. The streaming bitrate is the sum of core layer bitrate (fixed at 128kbps) and enhancement layer bitrate (from 0kbps onwards)

TABLE I
FILE SIZE COMPARISON IN BOTH LAYERS

Signal		Bossa	canzo	heart
Unwatermarked	Core	80k	96k	96k
	Enhancement	475k	359k	580k
Watermarked	Core	80k	96k	96k
	Enhancement	476k	361k	581k
Signal		Lasvega	nikuaile	slide
Unwatermarked	Core	81k	81k	80k
	Enhancement	515k	486k	495k
Watermarked	Core	81k	81k	80k
	Enhancement	516k	487k	496k
Signal		tears	toccata	young
Unwatermarked	Core	81k	81k	82k
	Enhancement	400k	539k	545k
Watermarked	Core	81k	81k	82k
	Enhancement	401k	540k	546k