ROBUST SPREAD-SPECTRUM AUDIO WATERMARKING

Darko Kirovski and Henrique Malvar

Microsoft Research, One Microsoft Way, Redmond, WA 98052

ABSTRACT

We present several mechanisms that enable effective spread-spectrum audio watermarking systems: prevention against detection desynchronization, cepstrum filtering, and chess watermarks. We have incorporated these techniques into a system capable of reliably detecting a watermark in an audio clip that has been modified using a composition of attacks that degrade the original audio characteristics well beyond the limit of acceptable quality. Such attacks include: fluctuating scaling in the time and frequency domain, compression, addition and multiplication of noise, resampling, requantization, normalization, filtering, and random cutting and pasting of signal samples.

1. INTRODUCTION

Traditional data protection techniques such as encryption are not adequate for audio copyright enforcement, because audio must be played back after decryption. Therefore, in most scenarios it is possible to record the decrypted content; in the worst case by recording the analog output of the playback device. By inserting watermarks in the audio content itself [1], one can enable copyright protection, while surviving the re-recording attack.

Most audio watermarking schemes rely on the imperfections of the human auditory system (HAS). In the time domain, it has been demonstrated that the HAS is insensitive to small level changes [2] and insertion of low-amplitude echoes [3]. Data hiding in the frequency domain takes advantage of the insensitivity of the HAS to small spectral magnitude changes [4]–[6]. Quantization index modulation is another type of data hiding algorithms that increases the security of the augmented data at the cost of decreased tolerance to attack noise stronger than the watermark modulation [7].

One of the most promising watermarking techniques relies on hiding a low-amplitude spread-spectrum (SS) sequence, which can be detected via correlation techniques [8]. Usually, embedding is performed in high amplitude portions of the signal [4]. In Section 2 we discuss the problems with traditional SS watermarking schemes. In Section 3 we introduce a novel, highly reliable and robust framework for hiding and detecting SS watermarks. By introducing new components to SS watermarking, such as chip redundancy, synchronization search, cepstrum filtering, and chess watermarks, we significantly improve detection performance. In Section 4 we discuss system implementation issues, and we show that our enhanced SS system is robust to a wide variety of signal manipulation attacks, including ones that degrade the original audio well beyond the limits of acceptable quality. Therefore, we believe our techniques can be integrated into an effective copyright enforcement system for distribution of high-fidelity digital music.

2. BASICS OF SS WATERMARKING

Let us denote as *x* the original signal vector to be watermarked. It represents a block of samples from an appropriate invertible transformation on the original audio signal [4], [6], [8]. The corresponding watermarked vector is generated by $y = x + w$, where the watermark *w* has elements w_i (chips) assigned to one of two equiprobable values, i.e. $w_i \in \{-\Delta, +\Delta\}$, independently of *x*. Parameter Δ should be set based on the sensitivity of the HAS to amplitude changes. In our case, *x* is a vector of magnitude frequency components in a decibel scale, so Δ should not be higher than about 1 dB. A correlation detector performs the optimal test for the presence of a watermark [8]:

$$C = y \cdot w = (x + w) \cdot w = x \cdot w + N\Delta^2 \qquad (1)$$

where *N* is the cardinality of the vectors, and the correlation between two vectors u and v is defined by $u \cdot v \triangleq \sum u_i v_i$. Under the mild assumption that the original clip *x* can be modeled as a Gaussian random vector, i.e. $x_i \sim \mathrm{N}(m_x, \sigma_x)$, $\sigma_x >> \Delta$, the value of the normalized correlation test is given by

$$Q \triangleq \frac{C}{N\Delta^2} = \rho + r \qquad (2)$$

where $\rho = 1$ if the watermark is present (and zero otherwise), and *r* is a correlation noise caused by the "carrier" signal *x*, with $r_i \sim \mathrm{N}(0, \sigma_r)$, $\sigma_r^2 = \sigma_x^2 \big/ \left(N\Delta^2\right)$. The optimal detection rule is to declare the watermark present if $Q > T$. The choice of the threshold *T* controls the tradeoff between false alarm and detection probabilities; e.g. if the watermark is absent, the false alarm probability is $\Pr[Q > T] = \mathrm{erfc}\left(T\sigma_x / \sqrt{N}\right)$ [9]. A similar analysis can be performed if *x* is assumed to be Laplacian, which is a better model if *x* is in a linear rather than decibel scale [8].

Advantages of SS watermarking include: (*i*) testing for watermarks does not require the original and (*ii*) watermark detection is exceptionally resilient to attacks that can be modeled as additive or multiplicative noise. Disadvantages include: (*i*) the watermarked signal and the watermark have to be perfectly synchronized while computing (1) and (*ii*) for a sufficiently small error probability, the vector length *N* may need to be quite large, increasing detection complexity and delay.

3. IMPROVING SS AUDIO WATERMARKING

In our audio watermarking system, the vector *x* is composed of the dB magnitudes of several frames of a modulated complex lapped transform (MCLT) [10]. After addition of the watermark,

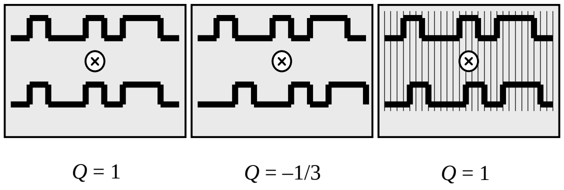

Figure 1. An example of using triple redundancy to improve the normalized correlation in the case of a desynchronization attack.

we generate the time-domain watermarked audio signal by combining the marked vector *y* with the original phase of *x*, and passing those modified frames through the inverse MCLT. For 44.1 kHz sampling, we use a length-2048 MCLT. Only the MCLT coefficients within the 2–7 kHz subband are modified and considered in the detection process, to minimize carrier noise effects as well as sensitivity to downsampling and compression.

3.1 Mechanisms Against Desynchronization Attacks

The correlation metric in (1) is reliable only if the detection chips w_i are aligned with those used in marking. Therefore, a malicious attacker can attempt to desynchronize the correlation by time- or frequency-scale modifications. We now describe a methodology for adding redundancy to the watermark chip pattern, so that the correlation metric is still reliable in the presence of scale modifications.

The basic idea behind redundant chip coding is shown in Figure 1. The leftmost subfigure depicts a perfect synchronization between a nine-chip watermark w and a nine-chip watermarked signal $y = x + w$ (assuming $x = 0$ in that segment). The normalized correlation in that case is $Q = 1$. However, if the watermark is shifted for one sample (middle of Figure 1), the normalized correlation becomes $Q = -1/3$, a major change. To prevent that sensitivity, we spread each chip of the SS sequence onto *R* consecutive samples, and during detection we include only chips at the center of the region. In our example, we use $R = 3$, which guarantees the desired result $Q = 1$ (right of Figure 1). In general, it is easy to show that the correlation is guaranteed to be correct even if a linear shift of up to floor(*R*/2) samples across the watermarking domain is induced.

In the example above, the frequency magnitude spectrum remains unchanged. However, when an audio track is played back at a different speed, MCLT components are shifted both in time and in frequency. Plus, the amount of shift is proportional to the center frequency and time of the MCLT coefficient position in the time-frequency plane. To maintain correlation reliability even under such conditions, we introduce chip redundancies both along the time and frequency axes. Each bit w_i of an SS sequence is spread (replicated):

- **in frequency**: over a subband of MCLT samples x_k that spans over $k \in F_j,\ j = 1, 2, \ldots, J$, consecutive frequency indices within a single MCLT frame (where *J* is the number of SS sequence bits per MCLT block),
- and **in time**: over T_0 consecutive MCLT frames.

The boundaries along the frequency axis $F_j,\ j = 1, 2, \ldots, J,$ are computed using a geometric progression:

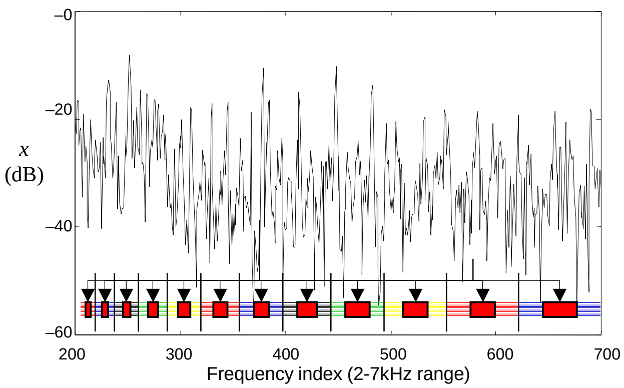

Figure 2. Illustration of geometrically progressed redundancies applied to SS sequence bits within a single freq-spectrum block. Each depicted subband is encoded with the same bit, whereas the detector integrates only the center locations of each region (indicated by the arrows).

$$V_F'\left(F_i'+\sum_{j=1}^{i-1}F_j\right)\le F_i',\qquad V_F'\left(F_i''+\delta_F+F_i'+\sum_{j=1}^{i-1}F_j\right)<F_i'',$$

where $F_i''+\delta_F+F_i'=F_i$ and δ_F is the width of the decoding region along the frequency axis and $V_F' > V_F$ is the desired variable frequency shift coverage. Similarly, the redundancy factor T_0 imposed along the time axis is replicated m times, where m is delimited by

$$mT_0V_T < T_0 - \delta_T \tag{3}$$

where δ_T is the width of the decoding region along the time axis and T_0 is the lower bound on the replication in the time domain (e.g. due to the impact of signal cropping or insertion of up to 100 ms). Once (3) cannot be satisfied, we iteratively compute T_k similarly to F_i. Within a region of F_jT_k samples watermarked with the same SS sequence bit, only the center $\delta_F\delta_T$ samples are integrated in the correlation test (1). It is straightforward to prove that such generation of encoding and decoding regions guarantees that regardless of induced V_F and V_T, the correlation test is performed in perfect synchronization. Figure 2 illustrates these concepts.

Resilience to static time and pitch scaling is obtained by performing multiple correlation tests. Each test assumes a different combination of time and pitch scaling. For example, in order to cover static time changes of ± 10% and static frequency changes of ± 5%, in steps of V_F' = 1%, the watermark detector needs to compute 105 different correlation tests. Note that V_F' has to be twice as large as V_F and that there should be a 50% overlap in coverage between two successive iterations. The search step along the time axis equals δ_T. Figure 3 depicts the normalized correlation *Q* values obtained from a detector during the watermark search for marked (top) and non-marked (bottom) audio clips. Peaks of *Q* values clearly indicate the existence and the location of each watermark.

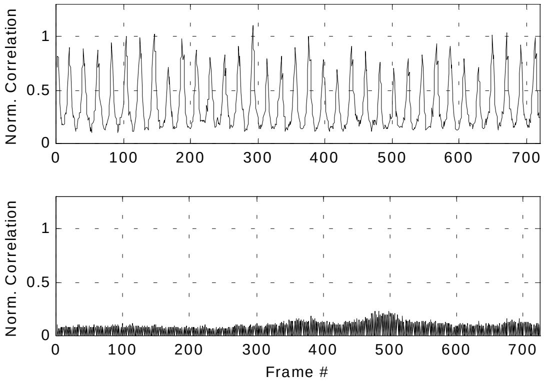

Figure 3. Maximum normalized correlation Q values retrieved during a search for a watermark in a marked (top) and unmarked (bottom) audio clip.

3.2 Cepstrum Filtering (CF) Pre-Processing

The variance σ_x^2 of the original signal directly affects the carrier noise r in (2). Audio clips with large energy fluctuations or with strong harmonics are especially bound to produce large σ_x. Thus we propose here a nonlinear processing step to reduce the carrier noise. One approach is to subtract a moving average from the frequency spectrum right before correlation; a sort of whitening step. Unfortunately, as bits of the SS sequence are spread over frequency ranges, this technique induces partial removal of the watermark chips. We have developed a cepstrum filtering (CF) technique that produces significantly better results than just spectral whitening.

With CF we reduce σ_x in (2) through the following steps:

- Compute an approximate cepstrum of the dB magnitude MCLT vector y under test via a discrete cosine transform (DCT) operator; $z = \mathrm{DCT}(y)$.
- Filter out the first K (typically $5 < K < 20$) cepstrum coefficients, i.e. set $z_i = 0,\ i = 0,1,...,K-1$.
- Reconstruct the frequency spectrum via an inverse DCT, $\tilde{y} = \mathrm{IDCT}(z)$. The filtered frequency spectrum $\tilde{y}$ replaces y in the correlation detector (1).

The rationale behind CF is that large variations in y can only come from large variations in x, since $|w|$ is limited to a small value Δ. Thus, by filtering out large variations in y we can reduce the carrier noise significantly, without affecting much the expected value ρ. That is particularly efficient if the watermark sequence w has a nonwhite spectrum containing more noise at higher frequencies, as discussed in the next subsection.

Figure 4 illustrates the impact of CF on the signal variance (top plot) and detection performance (middle and bottom plots). We see in the top plot that the signal variance is reduced by a factor of almost four. The detector in the middle plot does not use CF, whereas the one in the bottom plot does. All other parameters of the detection and embedding are equivalent. Thus, in order to attain the performance of CF detector, a non-CF detector must integrate almost four times more magnitude points.

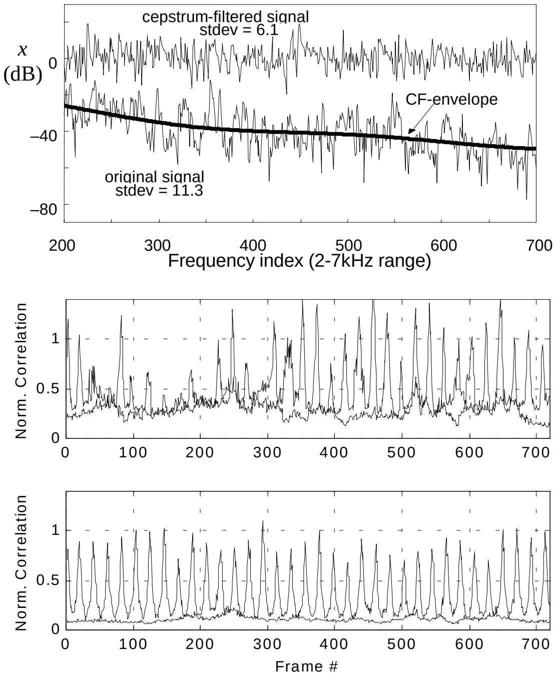

Figure 4. Variance reduction through cepstrum filtering (CF). Top: a typical watermarked signal $y = x + w$ prior to CF (the thick line is smooth envelope corresponding to the cepstrum coefficients that are filtered out), and the same y after CF. Middle: maximum normalized correlation Q values (for signals with and without watermarks, as in Figure 3) with CF turned off. Bottom: Q values with CF turned on; note the significant reduction in the variance of Q.

3.3 Chess Watermarks

Because of the relatively short MCLT frames (~30ms), we can assume that the audio signal has a slowly varying magnitude spectrum. Thus, for short watermarks, a possible sequence in time of consecutive watermark chips equal to +Δ can pose "false alarm" problems if correlated with large positive x values. In practice, that problem occurs frequently for quiet clips with strong harmonics (e.g. piano or sax solo). To alleviate the problem, it is important to attenuate the DC component of the watermarking chips along the time direction.

We define a "perfect watermark" (PW) as a sequence of alternating +Δ and –Δ chips (or 0 and 1 bits, respectively), along both the time and frequency axis. Correlation with PW results in highly improved correlation convergence for a non-watermarked signal, as illustrated in Figure 5 (a). To leverage the convergence efficacy of PW with the security of pseudo-random SS sequences, we introduce "chess-watermarks" (CW). We define a CW as a stochastic approximation to a PW, by using the simple first-order state machine depicted in Figure 5 (b). Whereas the probability p of switching from the 0 state to the 1 state for traditional SS sequences is desired to be one-half, we built CWs to enforce frequent toggling of bits along the time axis or, equivalently, to emphasize high frequencies in the watermark sequence.

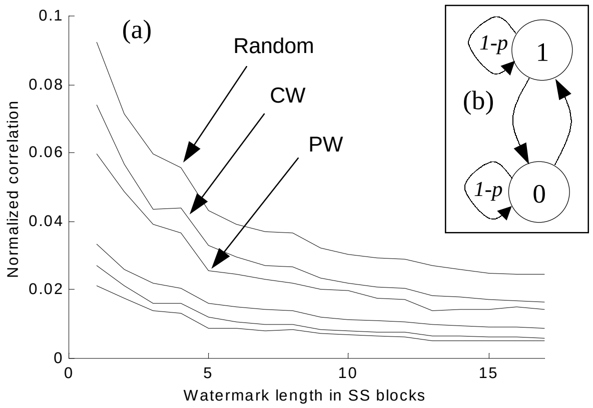

Figure 5. a) Normalized correlation *Q* convergence with watermark length, for a non-watermarked signal. Top three plots: 90% percentile limits of *Q* (90% of the correlation values are under each curve), for a traditional purely random SS sequence, a perfect watermark (PW), and a chess watermark (CW). Bottom three plots: the corresponding standard deviations of *Q*, in the same order. b) Simple state machine that produces a chess watermark ($p > 0.5$).

We typically select p around 0.75. In a typical implementation, we have enforced that consecutive 4-tuples of bits along the time axis are pseudo-randomly chosen from the following alphabet S = {0101,1010,1001,0110}, thus reducing the domain of all possible $4n$-bit SS sequences from 2^{4n} to 4^n. For a sufficiently large n, the reduction in the sequence domain does not pose a security threat, while resulting in correlation convergence similar to PW (typically $n > 200$).

4. TECHNOLOGY IMPLEMENTATION, ROBUSTNESS, AND SECURITY

We have designed a complete SS audio watermarking system using the enhancements described in the previous section. A reference implementation of our audio watermarking technology on an x86 platform requires about 32 KB of memory for code and 100 KB for data. The data buffer stores averaged frequency magnitudes for 12.1 seconds of audio, for a watermark length of 11 s. Watermarks are searched with $V_F' = \pm 2\%$, which requires ~ 40 tests per search point. Real-time watermark detection under these circumstances requires ~ 15 MIPS. Watermark encoding is an order of magnitude faster, with smaller memory footprints.

While watermarking techniques for images can be tested with the Stirmark tool [11], to date a similar benchmark has not been developed for audio. Thus, we have tested our proposed watermarking technology using a composition of common sound editing tools and malicious attacks [12], including all tests defined by the Secure Digital Music Initiative (SDMI) [13]. We tested the system against a benchmark suite of eighty 15-sec audio clips, which included: jazz, classical, voice, pop, instrument solos (accordion, piano, guitar, sax, etc.), and rock. In that dataset, there were no errors, and we estimated the error probability to be well below 10^{-6}. Significantly lower error probabilities can be achieved by increasing the watermark length.

Other attacks against SS audio watermarking include: (*i*) desynchronization (discussed in Subsection 3.1), (*ii*) averaging – which we prevent by placing watermarks at random positions in an audio clip, and (*iii*) exhaustive search of watermark bits by adding noise to the audio signal– which we address by adding a pseudo-random biased offset to the threshold T at each test.

5. CONCLUSION

A common deficiency in SS watermarking systems, which perform correlation-based detection, is lack of robustness against desynchronization and large amplitude variations in the audio carrier. We have developed a set of novel techniques that significantly improve SS watermark systems, by imposing particular structures to watermark patterns (while retaining plenty of randomness) and applying a nonlinear filter to reduce carrier noise. Using these techniques, we developed an audio protection system that is capable to reliably detect watermarks, even in audio clips that have been modified using a composition of attacks that degrade the content well beyond the limit of acceptable quality.

6. REFERENCES

[1] S. Katzenbeisser and F. A. P. Petitcolas, Eds. Information Hiding Techniques for Steganography and Digital Watermarking. Boston, MA: Artech House, 2000.

[2] P. Bassia and I. Pitas, “Robust audio watermarking in the time domain,” *Proc. EUSIPCO 98,* vol. 1, pp. 25–28, Rodos, Greece, Sept. 1998.

[3] D. Gruhl, A. Lu, and W. Bender, “Echo hiding,” in *Information Hiding,* Springer Lecture Notes in Computer Science, vol. 1174, pp. 295–315, 1996.

[4] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, “A secure, robust watermark for multimedia,” Information Hiding Workshop, Univ. of Cambridge, pp.185–206, 1996.

[5] C. Neubauer and J. Herre, “Digital watermarking and its influence on audio quality,” *Proc. 105th Convention,* Audio Engineering Society, San Francisco, CA, Sept. 1998.

[6] M.D. Swanson, B. Zhu, A.H. Tewfik, and L. Boney, “Robust audio watermarking using perceptual masking,” *Signal Processing,* vol.66, pp. 337–355, 1998.

[7] B. Chen and G. W. Wornell, “Digital watermarking and Information embedding using dither modulation,” *Proc. IEEE Workshop on Multimedia Signal Processing,* Redondo Beach, CA, pp. 273–278, Dec. 1998.

[8] W. Szepanski, “A signal theoretic method for creating forgery-proof documents for automatic verification,” *Proc. Carnahan Conf. on Crime Countermeasures*, Lexington, KY, pp. 101–109, May 1979.

[9] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I.* New York: Wiley, 1968.

[10] H. S. Malvar, “A modulated complex lapped transform and its application to audio processing,” *Proc. ICASSP,* Phoenix, AZ, pp. 1421–1424, 1999.

[11] R. J. Anderson and F. A. P. Petitcolas, “On the limits of steganography,” *IEEE J. Selected Areas in Communications,* vol.16, pp. 474–481, 1998.

[12] F. Hartung, J. K. Su, and B. Girod, “Spread spectrum watermarking: malicious attacks and counter-attacks,” *Proc. SPIE Security and Watermarking of Multimedia Contents,* San Jose, CA, pp. 147–158, Jan. 1999.

[13] Call for Proposals, Phase I, http://www.sdmi.org, 1999.