Comparison of Audio Watermarking-Techniques

Master Hauptseminar Medientechnologie WS 15/16

Stephan Wiefling
Technische Hochschule Köln
stephan.wiefling@smail.th-koeln.de

ABSTRACT

Audio watermarking is a widely used technology to hide information about the reciever of an audiofile inside the time- or spectral components of the original audiosignal, with the aim of being imperceptible to the human auditory system. If the watermarked audiofile appears illegal on the Internet, the unauthorized circulator of the audiomaterial can be ascertained by decoding the watermark back from the signal-components.

In this paper, several techniques of audio watermarking are presented with their different approaches of hiding information in the audio-data and evaluated in terms of robustness against audio-manipulations and signal suitability. Furthermore, the current status of the audio-watermarking-research will be shown.

Keywords

Audio watermarking, signal processing

1. INTRODUCTION

With the spread of digital technology, enormous amounts of information can be accessed, in the appearance of images, text and audio [6]. Audio compression technologies like MPEG-2 Audio Layer III (MP3) or Advanced Audio Coding (AAC) enables the transfer of audio information with a slight amount of data, compared to the uncompressed raw data. Nevertheless, these technologies also provide opportunities for people to spread illegal copies of protected works over the Internet [20]. Therefore, several methods of protecting copyright ownership have been developed.

Watermarking, belonging to the field of Data Hiding [6], is a technology embedding *"data into digital media for the purpose of identification, annotation and copyright"* [4]. First developed for the use in digital images (e.g. [31]), several Watermarking techniques for digital audio data have been proposed afterwards (e.g. [4, 6]). In contrast to the field of cryptography, the protected audio content itself is not encrypted and the receiver of a watermarked audio signal may not know or even perceive the presence of hidden information [22].

Starting with the first developed algorithms in the 1990s (e.g. [10]), audio watermarking has gained in importance to date. Matt Montag, an employee of the internet audio streaming service Spotify, pointed out with an online listening test[1] conducted at his homepage, that the music corporation Universal Music Group (UMG) utilizes audio watermarking in works of their artists, with audible artifacts to some extent, on his employer Spotify and also e.g. on the online music store iTunes or inside broadcasts of UMG songs over FM radio[2]. Moreover, classical music enthusiasts on the audio internet forum Hydrogenaudio[3] noticed watermark artifacts inside lossless audiofiles by UMG, purchased in an online music store for lossless audio. Examples like these underline, that audio watermarking is a topic with high relevancy in today's digital world.

As many researchers in this topic have exhibited, algorithms in audio watermarking should meet the following requirements:

1. Imperceptibility, respectively minimal perceptibility, of the embedded signal by the human auditory system (HAS) while preserving the perceptional audio quality of the host signal [4, 6, 30, 32, 5, 1].

2. Embedding of the watermark directly into the audio data and not in the audio header or wrapper, respectively [6, 4].

3. Robustness against any form of audio data manipulation or processing with the intend of removing or corrupting the embedded watermark signal, e.g. lossy audio data compression, addition of noise, filtering [5,

[1] M. Montag. Watermark Listening Test Results, January 2015. http://www.mattmontag.com/music/listening-test-results

[2] M. Montag. Universal's Audible Watermark, January 2012. http://www.mattmontag.com/music/universals-audible-watermark

[3] Universal MG embed an audible watermark in downloads. July 2011. https://www.hydrogenaud.io/forums/index.php?showtopic=89818

4, 6, 28, 1]. Successful removal of the watermark signal should only be achievable with massive auditory quality degradation of the host signal as a result [12].

Beyond that, several papers suggest that the watermark data should inhere the ability of being self-clocking [6, 4, 1] or support the addition of multiple watermarks [6] for enhanced robustness and also include error correction coding to ensure data accuracy [4]. All the referred algorithms in this paper are based on the embedding of the binary states "one" or "zero" (watermark bits). Algorithms in audio watermarking can either be dependent on the original signal (non-blind) or independent (blind) for watermark detection [11].

This paper considers different types of audio watermarking techniques and shows their different approaches in integrating watermark data directly into the audio data. Moreover, the techniques are evaluated in terms of robustness and suitability for different types of audio signals. Since most watermarking techniques hide their information in the time domain of the host signal [21], several techniques of that category are discussed in this paper.

The presented audio watermarking techniques are classified into the three categories *spread spectrum*, *echo hiding* and *low frequency based watermarking*. The selection of these categories is based on the number of publications covering the particular watermarking approach and the number of outside references towards these papers, underlining their scientific relevancy. Also the presented watermarking method should inhere a minimum amount of robustness against signal manipulation attempts, hence the often referenced *low-bit coding* watermarking method [4], where information is embedded by altering the least significant bit of an audio sample, is not covered. Other audio watermarking techniques include *phase coding* [4], *modified patchwork algorithm* [37], *cepstrum domain* [19] and *histogram based watermarking* [33]. Due to the limited page space, they could not be discussed here in detail.

The scientifically first presented and commonly used [19] method, *spread spectrum* watermarking, is presented first, followed by the *echo hiding* approach, *low frequency watermarking* and topics of current watermarking research.

2. SPREAD SPECTRUM

The spread spectrum watermarking technique was first introduced by Cox et al. [8]. Even though the paper showed watermarking experiments on digital images, the general approach could also be applicable for digital audio data [8]. It is based on the basic theory of direct sequence spread spectrum (DSSS), where a transmitted signal is spread as large across the frequency spectrum as possible by the multiplication with a code, which is independent of the transmitted signal [27, 4]. The basic principle of DSSS watermarking in the specific domain of audio data was first presented by Bender et al. [4] and is described in the following section.

2.1 Basic principle

To encode a watermark in the presented method, four signals are required: The original signal, the carrier wave, the key signal ("Chip") and the binary string. The carrier wave is a sinusoidal wave of a predefined frequency, the key signal consists of a pseudorandom noise (PN) sequence, whereas the binary string contains a watermark bit string for encoding. The last two signals only comprise two discreet amplitude values (-1, +1). The key signal is required to decode and encode the binary string and has the data rate of the original signal.

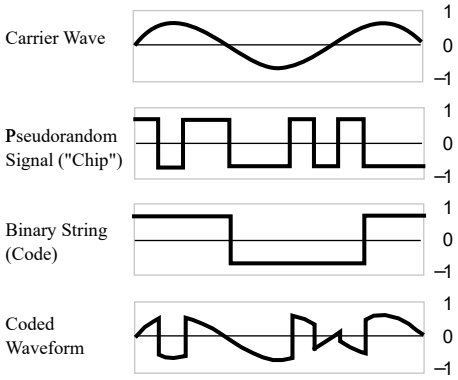


Figure 1: Watermark encoding with the DSSS method (taken from [4]): The carrier wave is multiplied with the pseudorandom key and the watermark code, resulting in the coded waveform (watermark signal)

For the embedding process, the amplitudes of the carrier wave are first multiplied in the time domain by the key and the binary string sample by sample. As a result, the spectrum of the resulting coded waveform is spread over a wide frequency range and has the characteristics of additive random noise. The center of the coded waveform's frequency range is defined by the frequency of the carrier wave. A schematic example of the synthesizing process can be seen in Figure 1. The coded waveform (watermark signal) is then attenuated at around 0.5 of the original data's dynamic range and added to the original data. The result is the watermarked signal.

For the detection process it has to be assured, that the following conditions are fulfilled: The key signal does not repeat for a long time range, has a flat frequency spectrum and is known for the decoder, the signals are synchronized in the time domain and also the beginning and the end of the DSSS-data inside the watermarked signal is known. Beyond that, the data rate of the key and the binary string, as well as the carrier frequency have to be known. Since the phase of the coded waveform alternates with every alternation of key and watermark, binary phase shift keying (BPSK) is used for the decoding process. Hence, the phase value ϕ is interpreted as "0" and $\phi + \pi$ as a "1", resulting in the encoded watermark.

As Bender et al. pointed out, the described method leaves room for improvements. While the method led to *"some degree of success"*, it could not assure protection against possible watermark removal attempts. Furthermore, it did not include methods of adaptive watermark coding in terms of compressed audio material and protection methods against

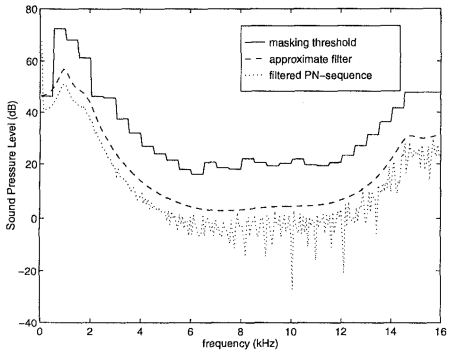


Figure 2: Filtered PN-Sequence (taken from [6])

signal modification attempts and possible interactions between different watermarking techniques.

An enhancement of the here described basic principle was presented by Boney et al. [6] afterwards. Their approach is described in the following section.

2.2 Improvement with perceptual masking

The effect of perceptual masking refers to the phenomenon, where a sound becomes inaudible by the presence of another louder sound [25, 22]. The masking effect can be accomplished by two sounds with close frequencies, which are presented simultaneously (simultaneous masking) or consecutively (temporal masking) [25, 22]. This characteristic of the human ear is exploited in audio compression technologies like MP3 or AAC, which is why it has been implemented in spread spectrum audio watermarking algorithms for additional robustness.

The DSSS watermarking method by Boney et al. [6] extends the previous described method by using perceptual masking models of the HAS for frequency shaping the watermark signal. It was slightly altered later by Swanson et al. [30], with two authors of [6] involved. Similar approaches were also presented by Seok and Hong [28] and Cvejic et al. [9]. This section is a combination of all four versions. Since different notations for equivalent terms were used in [6] and [30], they were uniformly adapted to one notation.

At the beginning of the watermark embedding process, a PN sequence, used as the watermark signal in these implementations, is generated. This sequence can be long or changing with each watermark block to avoid plain statistical detectability of the watermark by an unauthorized person. The PN generation process differs in all presented implementations: While a shift register was used for PN generation in [6], a PN generation progress using two random keys (e.g. RSA) was applied in [30]. The implementation in [9] uses an m-sequence for generation. The exact generation process in [28] is not mentioned by the authors.

For each block of 512 [6, 30, 28] or 1023 [9] consecutive audio samples of the original signal, the masking threshold in the frequency domain is calculated using the masking model specified in the ISO-MPEG Audio Psychoacoustic Model 1 for Layer I (see Figure 2). In combination with the original signal, audio signals, with frequency characteristics falling below the determined masking thresholds, are imperceptible to the HAS. The masking threshold is approximated with an all-pole filter (in [6]) and then applied on the PN sequence, whose frequency spectrum is then below the calculated masking threshold (see Figure 2).

However, signal impulses shorter than the block size may lead to audible distortions like pre-echoes, since the frequency calculations are based on Fourier transformations, which require a fixed window length in this case. To consider these supplemental temporal masking effects, the watermark signal is modified with a time weighting function, approximated by the envelope of the original signal in the time domain. Modulated with the original signal, the watermark signal of the first stage $watermark_{firststage}$ is generated. Contrary to that, a time weighting function is not mentioned in [28].

At this point, the final watermark is generated in [30], [28] and [9]. For additional robustness against lossy compression, according to the authors, a low frequency watermark w_{br} is added in [6]. However, in [30] it is no longer present. It is described as the difference between the watermark signal of the first stage and the original signal, while both signals are encoded at a low bitrate br (64 kbit/s for 44,1kHz sampling frequency) with a *"low bitrate audio coding algorithm"* [6]:

$$w_{br} = (watermark_{firststage})_{br} - (originalsignal)_{br} \quad (1)$$

The authors also added watermark information in the high frequency bands to improve the watermark detection at high bit rates. This can be done by creating the watermark w_{err} for the coding error, which is the difference between the original signal and the same signal encoded at a low bitrate:

$$codingerror = (originalsignal) - (originalsignal)_{br} \quad (2)$$

The final watermark w for [6] is generated by the addition of the low frequency watermark and the watermark for the coding error:

$$w = w_{br} + w_{err} \quad (3)$$

With the addition of the original signal $s(k)$ and the watermark signal w, the watermarked signal is generated.

Since the detection process in [6, 30] elementary differs from [28, 9], it is described first. For decoding the watermark in [6, 30], access to the original signal as well as the PN sequence used for embedding has to be ensured in both [6] and [30]. Beyond that, regarding the additional low frequency watermark, the *"approximate [MPEG coding algorithm] bit rate of the observed audio sequence"* has to be known in [6]. To estimate whether the observed audio sequence has been watermarked or not, the given signal $r(k)$ is substracted from a MPEG coded version of the original signal in [6], respectively the original signal $s(k)$ in [30], leading to the following hypothesis test problem:

$$\begin{aligned} H_0 &: x(k) = r(k) - s(k) = n(k) \\ H_1 &: x(k) = r(k) - s(k) = w'(k) + n(k) \end{aligned} \quad (4)$$

If condition H_1 is met, a watermark is included in $r(k)$, whereas in H_0, the opposite is fulfilled. $n(k)$ is the additive noise resulting from errors due different audio coding algorithms as well as transmission noise and jamming signals

(e.g. from intentional attacks on the watermark) and $w'(k)$ is the potentially altered watermark signal. Solving the hypothesis problem is mastered by correlating the similarity between $x(k)$ and the original watermark $w(k)$

$$Sim(x,w) = \frac{\sum_{j=0}^{N-1} x(j)w(j)}{\sum_{j=0}^{N-1} w(j)w(j)} \quad (5)$$

and comparing the result with a predefined threshold. At this point the watermark test in [6] is finished, whereas [30] further analyzes $r(k)$ in the case of an unknown location inside $s(k)$ with additional disturbance $d(k)$. It is assumed that

$$r(k) = s(k+\tau) + d(k) \quad (6)$$

with τ being the unknown delay, which is not necessarily an integer. If the generalized likelihood ratio test

$$\frac{max_\tau exp(-\sum_{k=0}^{N-1}(r(k) - (s(k+\tau) + w(k+\tau)))^2)}{max_\tau exp(-\sum_{k=0}^{N-1}(r(k) - s(k+\tau))^2)} \quad (7)$$

is higher than a predefined threshold, the authors assume a present watermark.

In opposite to [6, 30], the detection process in [28, 9] is blind, ergo no access to the original signal is required. To improve the detection performance, according to the authors, the audio spectrum of the given watermarked signal is first equalized by applying de-correlation or whitening with linear predictive coding (LPC) [2]. The exact equalization method is not mentioned in [9]. This process is also applied on the PN sequence used for embedding. By applying matched filtering on the PN sequence and the watermarked signal, the watermark is detected in [28]. The watermark in [9] is detected by cross-correlating the PN sequence with the watermarked signal and comparing the result against a certain threshold for the bit decision.

The DSSS watermarking methods using a psychoacoustic masking model have since been enhanced with method aiming at a higher robustness against intentional watermark removal attacks. Some approaches are discussed next.

2.3 Further enhancements

Kirovski and Malvar [14, 16, 15] refined DSSS watermarking techniques in the sense of supplementary robustness of the watermark. The embedding process in the paper is similar to the described methods before, while the decoding process is based on performing a correlation test between the given PN sequence and the watermarked signal. If the resulting correlation is above a predefined threshold, then the corresponding PN sequence is present in the watermarked signal.

As a method for preventing intentional desynchronization attacks, they introduced block repetition coding for adding redundancy into the time and frequency components of the watermark. Considering the watermarked signal is shifted in time and/or frequency domain by an assumed attacker, the decoder's originally correlation check Q could return inaccurate results, since this method requires alignment of most PN sequence's samples to the watermarked signal's samples [15], as it can be seen in the middle of Figure 3. By spreading each watermark bit of the PN sequence in time onto three

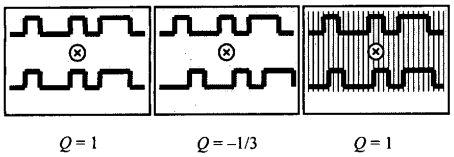


Figure 3: Example usage of the block repetition coding check (right) for improvement of the normalized correlation check Q in the case of desynchronization attempts (taken from [14])

consecutive samples in the time domain and only including the center of this region, ergo one of three samples, in the detection process, the watermark can be detected again, as we can see on the right area of Figure 3. In addition, each watermark bit is spread in frequency over three subbands. As similar as in the time domain, only the center of each dedicated frequency region is integrated for decoding.

In addition to that, the detector algorithm in [16, 15] performs multiple correlation tests over the corresponding blocks, each containing three time and frequency blocks. Inside each block, a smaller block in the size of the watermark length is taken and correlation tested with different scalings of the watermark. The smaller block with the highest correlation value is then compared against a threshold for determining the watermark presence. If a watermark is found, the next block is analyzed in the same way. If the opposite is true, the next smaller block is analyzed. By performing these steps, a higher robustness of the presented watermarking technique may be achieved. More details on the general robustness are discussed next.

2.4 Robustness and signal suitability

DSSS watermarking with perceptual masking is suitable for different types of audio signals. Tests conducted with a classical piano music composition by Schubert, the acapella song "Tom's Diner" by Suzanne Vega, a castagnet and a clarinet signal [6, 30], as well as six unknown audio pieces [28] showed a high watermark recovery rate.

Test results also showed a high robustness to MP3 coding with the lowest tested bitrate being 64 kb/s [6, 30, 28] and additional possible detection of audio data containing multiple watermarks [6, 30]. In addition, the proposed schemes are robust against signal manipulations on the watermarked signal, such as bandpass/lowpass filtering [28, 30] A/D and D/A conversion, echo addition, amplitude compression [28], additive colored noise, random cropping and resampling [30].

Nevertheless, DSSS watermarking methods have its vulnerabilities against any form of time scale modifications, respectively *"time-shifts or frequency scalings"* [5]. This suggests a well synchronized decoding algorithm for solving this task [5]. Kirovski and Malvar's implementation may inhere these requirements: Proceeded tests with the Stirmark Audio Benchmark [29], a test tool containing an extensive set of audio signal manipulation attacks, showed high recovery results for almost every test scenario [16]. The only attack siginificantly reducing the watermark *"had a strong impact on the fidelity of the recording so that the attacked clip almost did not resemble the original"* [16].

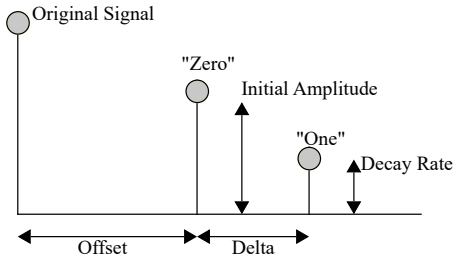


Figure 4: Adjustable parameters for the single echo method (taken from [4])

Nevertheless, this method requires more space in the time and frequency domain for watermark embedding, compared to the other presented DSSS watermarking methods. Kirovski and Malvar's robust method achieves a watermark bit rate of 0.5-1 bit/s [16], while other presented methods have an embedding capacity of 4 bit/s (Bender et al. [4]) or 14.7 bit/s (Cvejic et al. [9]). Note that the embedding bit rates of the remaining algorithms are not mentioned in the corresponding papers.

Besides the spread spectrum approach, watermark data can as well be embedded in the method of echo hiding, which is discussed in the following abstract.

3. ECHO HIDING

The echo hiding watermarking method was first introduced by Gruhl et al. [10] and embeds data bits into the time domain of the original audio signal by introducing an echo. The basic principle, the single echo method, is described in the following section.

3.1 Single echo method

The single echo method by Gruhl et al. [10] introduces a single echo with a positive amplitude for each watermark bit. This section is a combination of [4, 10].

The type of the embedded watermark bit is defined by two distinct delay times, representing either a "1" or a "0". Thus, for embedding binary data into the original audio signal, two system functions (kernels) are used, each containing two impulses from discreet time exponentials (see Figure 5). By convolving one of the kernels with the original signal, the first impulse copies the original signal, while the second impulse creates an echo. For each kernel, different echo delay times and amplitudes are used (see Figure 4): One for representing a binary zero (offset with initial amplitude) and one for representing a binary zero (offset+delta with decay rate). With decreasing value of the offset, ergo the distance between the original signal and the echo, the echo is percieved as added resonance to the HAS. The delay times δ_0, δ_1 are chosen below the HAS threshold of distinguishing between the echo and the original signal in the time domain. In addition, the echo amplitudes are below the hearing threshold of the HAS.

For the watermark data embedding process, the original audio signal is divided into smaller sections in the time domain. By convolving each section with either kernel "zero" (see Figure 7(b)) or kernel "one" (see Figure 7(a)), a binary

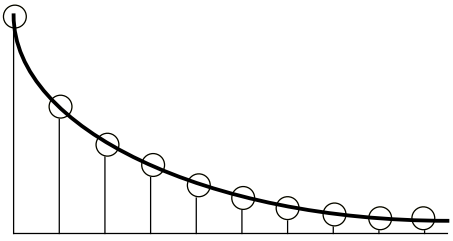


Figure 5: Example for a discreet time exponential (taken from [4])

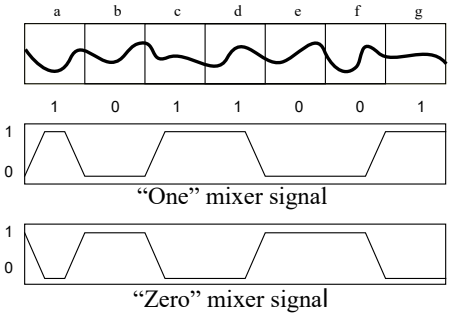


Figure 6: Mixer signals (taken from [4])

zero or a binary one is embedded in the corresponding section. To achieve smooth transitions between the echoes and a therefore "less noticeable mix", two echo signals are created. Echo signal "one" contains the entire original signal convolved with kernel "one" and echo signal "zero" contains the entire original signal convolved with kernel "zero". Also two mixer signals for the two echo signals, "one" and "zero", are created, each containing the values 1 ("encode the bit") or 0 ("do not encode the bit"). Mixer signal "one" is multipied with the echo signal "one", while mixer signal "zero" is multiplied with the echo signal "zero". Transitions between 1 and 0 and vice versa contain a linear rise on the one mixer signal and a linear decline on the other mixer signal, resulting in a "crossfade" between both echo signals. The procedure is pictured in Figure 6. With the combination of both mixed signals, the watermarked audio signal is created.

For decoding, the watermarked audio signal is divided into smaller sections in the time domain, in the same manner as in the embedding process. To decode the watermark data, one has to detect the spacing between the echoes inside the watermarked audio signal. In order to do this, the absolute amplitude of the watermark signal's autocepstrum at the two supposed echo positions have to be compared. The autocepstrum is described as the autocorrelation of the signal's complex cepstrum, which is an *"echo detection method"* [13] and defined as *"inverse Fourier transform of the logarithm of the Fourier transform of that function"* [13].

The capability of the cepstrum for echo detection is briefly reasoned as follows (derived form [13], [35] and [24], adapted to one notation): Let the echo kernel be the signal

$$x(t) = s(t) + a * s(t - t_0) \tag{8}$$

with $s(t)$ being the first impulse and $s(t - t_0)$ being the echo impulse, multiplied by the amplitude scaling factor a.

With Fourier transforming $x(t)$ to the frequency domain, the spectrum yields

$$X(\omega) = S(\omega) * (1 + a * \epsilon^{-j\omega t_0}) \tag{9}$$

where $X(\omega)$ and $S(\omega)$ are the Fourier transformations of $x(t)$ and $s(t)$, respectively. Taking the logarithm

$$log(X(\omega)) = log(S(\omega)) + log(1 + a * \epsilon^{-j\omega t_0}) \tag{10}$$

will alter the multiplication into an addition. Knowing that

$$log(1 - x) = x - \frac{x^2}{2} + \frac{x^3}{3} - ... \tag{11}$$

inverse Fourier transforming it back into the time domain will result in the cepstrum:

$$\begin{aligned} F^{-1}\{log(X(\omega))\} = F^{-1}\{log(S(\omega))\} + \\ a_0 * \delta(t - t_0) - \frac{a_0^2}{2} * \delta(t - 2t_0 + ... \end{aligned} \tag{12}$$

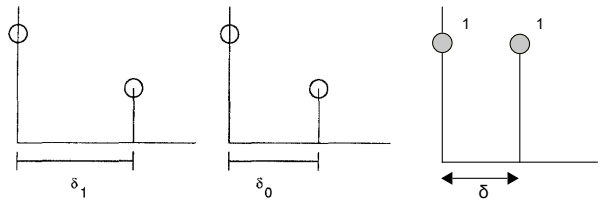
Looking at Equation 12, it is obvious that the signal peaks δ in the cepstrum repeat periodically at multiples of the delay position t_0, proving the echo detection ability of the cepstrum.

Due to the fact that the amplitudes the embedded echoes are comparably small in contrast to the original signal, the echo delays are hard to determine in the watermarked signal's autocepstrum. A solution to this problem is echoing of the watermarked signal with an echo kernel (see Figure 8). Assuming that the echo kernel's delay is shorter than the previously embedded echo delays, only the original signal is amplified (see 8(b)). As a result, embedded echoes are also strengthened by following impulses, visible as spikes in the autocepstrum (see Figure 9). If the amplitude in the autocepstrum of one small section is higher at δ_0 seconds than it is at δ_1 seconds, then a "0" is assigned for the corresponding bit position of the section. A "1" is assigned, if the opposite is true. If all portions are analyzed, then the embedded watermark bits are decoded. If the length of the embedded watermark bit sequence is known, the rule of majority can lead to the final decision of the then decoded watermark bit sequence.

However, the simple approach has since been enhanced by later works using distinct echo kernels. Several echo hiding methods with multiple echoes are depicted in the succeeding section.

3.2 Multi echo methods

Oh et al. [23], Ko et al. [17, 18] and Xu et al. [36] enhanced the basic echo hiding method with different watermarking approaches containing multiple echoes for decoding watermark information.



(a) "one" Kernel (b) "zero" kernel (c) echo kernel

Figure 7: Overview of Kernels (taken from [10, 4])

Oh et al.'s method [23] proposed an echo kernel design containing multiple positive and negative echo impulses with different and *"closely located"* delay times. As a result, low frequency components are attenuated in the frequency response of the echo kernel (see Figure 10), resulting in a *"more transparent"* sounding result of the watermarked audio signal [23], compared to the previously presented echo hiding methods. Decoding of the watermark is achieved by detecting the signal peaks in the watermarked signal's autocepstrum at the possible echo delay positions, similar to the previously described echo detection method in section 3.1.

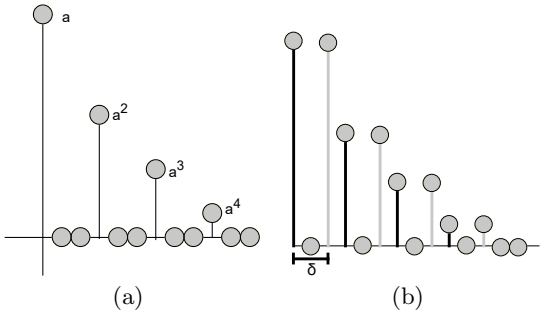
The time spread echo approach by Ko et al. [17, 18] applies an echo kernel containing a single echo, which is spread in time using a PN sequence after a delay $\triangle$ (see Figure 11). The PN sequence is containing the watermark bit information. Thus, in contrast to other echo hiding methods, only one echo kernel is used for encoding the entire watermark information. For decoding the watermark information of the watermarked audio signal, the PN sequence used for encoding has to be known to the decoder. In addition, the echo has to be despreaded in the time domain. By cross-correlating the cepstrum of the watermarked signal $c(k)$ with the used PN sequence $PN(k)$

$$d(k) = xcorr(c, PN) \tag{13}$$

the embedded watermark can be detected by investigating a peak at the delay position in $d(k)$, in other words a peak at $d(\triangle)$.

In contrast to embedding one watermark bit with one large echo, Xu et al.'s echo hopping technique encodes one watermark bit with four smaller consecutive echoes, each with a different time delay [36]. The echo delays representing the binary values are adaptively chosen in dependency of the presented audio signal type. Decoding the watermark is achieved by segmenting the watermarked audio with the same method used in the encoding process and detecting the four echo delays inside each segment.

Further details on the robustness and signal suitability properties of the presented echo hiding methods is discussed in the following abstract.



(a) (b)

Figure 8: a: Example signal, b: Example signal convolved with the echo kernel (Figure 7(c)) (taken from [4])

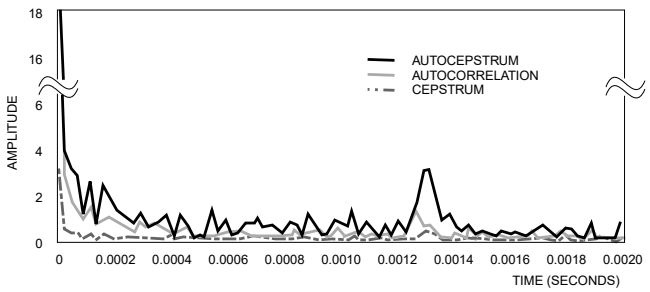
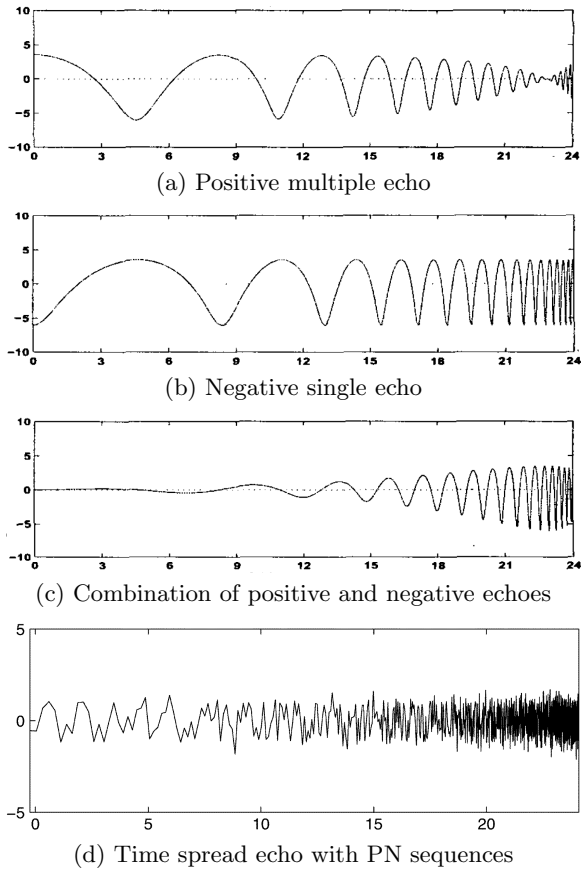


Figure 9: Example result of autocorrelation, cepstrum and autocepstum for a "zero" bit, after convolving the watermarked signal with the echo kernel (Figure 7(c)) (taken from [4])



(a) Positive multiple echo

(b) Negative single echo

(c) Combination of positive and negative echoes

(d) Time spread echo with PN sequences

Figure 10: Frequency response for different echo kernels (taken from [23, 18]). X-axis: Critical-band rate [Bark], y-axis: Magnitude response [dB]

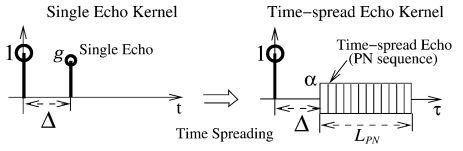


Figure 11: Single echo kernel and time spread echo kernel using a PN sequence (taken from [18])

3.3 Robustness and signal suitability

Echo hiding methods are not suitable for all types of audio signals. Since the single echo method results in possible addition of resonance and therefore alteration of the audio signal perception, it might *"be problematic in some music applications"* [4]. Note that in some applications this might also give *"the signal a slightly richer sound"*, according to Gruhl et al. [4]. In General, the echo hiding method has its difficulties in the watermarking of signals with large parts of total silence [10], since no echo may be encoded in this area. In tests conducted by Gruhl et al., a segment of a popular music signal showed more accurate decoding results than a spoken word signal [10]. But altogether, with 20 distinct audio signals tested, all cases showed an *"acceptable accuracy"* with watermark recovery rates above 85% [10]. However, similar to spread spectrum watermarking methods, the time spread echo method is not robust against pitch scale modifications [18].

Also the robustness against MPEG audio compression is signal dependent: While the MPEG decoded popular music signal showed no significant decline in the watermark recovery rate, MPEG decoded spoken word signals did in [10]. Note that neither details about the exact codec, except being *"MPEG"*, nor the exact bitrate used in the recovery tests are mentioned in [10]. On the other hand, tests using the multi echo method by Oh et al. and the time spread echo method showed comparably low bit error rates, respectively good watermark recovery results for MP3 coding with a bitrate of 56 kb/s [23] or 128 kb/s [18], noting that the details about the tested audio signal types are not mentioned in both papers.

A *"typical"* data embedding rate value for Echo Hiding is 16 bit/s in [10]. After spread spectrum and echo hiding, another possible technique of audio watermarking is presented next.

4. LOW FREQUENCY BASED WATERMARKING

Bassia et al. [3] and Lie and Chang [21] proposed different audio watermarking algorithms with the similarity of embedding the watermark data inside the low frequency domain of the original signal. Since both methods are frequently referenced in literature, they are described in this section.

The approach of Bassia et al. [3] is partially similar to spread spectrum watermarking: The original signal is partitioned into segments containing N samples. It generates a PN sequence with the length of N containing the values -1 or +1, generated by a chaotic map, used as the watermark key. For each segment, this PN sequence is than attenuated with a predefined factor and modulated with the original audio signal's segment. In contrast to DSSS, the result is then low pass filtered with a Hamming filter, resulting in an *"inaudible watermark"* [3], if combined with the original signal. The filtered result is added to the original signal segment. Once this procedure is repeated for every segment, the watermarked signal is generated.

For detection, the watermarked signal is segmented into

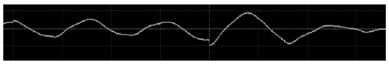


Figure 12: Signal discontinuities in the time domain while using Low Frequency Amplitude Modification watermarking without progressive scaling (taken from [21])

frames containing N samples. In addition, the PN sequence used for the watermark has to be known to the decoder. With correlation checks between the watermark and the watermarked signal for all possible circular shifts $S_k(n)$ and the correlation between the original signal and the watermark $T_{2,k}(n)$, the presence of a watermark is determined with the ratio

$$r_k(n) \hat{=} \frac{S_k(n) - T_{2,k}(n)}{T_{3,k(n)}} \tag{14}$$

with $T_{3,k(n)}$ being the watermarked signal. If the maximum value of $r_k(n)$ for all frames is above a predefined threshold, the watermark is detected.

Lie and Chang's watermarking method [21] is based on relative relations between consecutive audio frames, resulting in a low frequency amplitude modification. For embedding the watermark bit sequence, the original signal is divided into consecutive groups of samples (GOSs), each divided into three consecutive sections L_1, L_2, L_3 of equal or unequal length. One GOS consisting of $L = L_1 + L_2 + L_3$ sections is representing one watermark bit value. The embedding of the watermark bit value inside each GOS is depending on the average of absolute amplitudes (AOAA) of each section, defined as

$$E_{i1} = \frac{1}{L_1} * \sum_{x=0}^{L_1-1} |f(L * i + x) \tag{15}$$

$$E_{i2} = \frac{1}{L_2} * \sum_{x=0}^{L_2-1} |f(L * i + x) \tag{16}$$

$$E_{i3} = \frac{1}{L_3} * \sum_{x=0}^{L_3-1} |f(L * i + x) \tag{17}$$

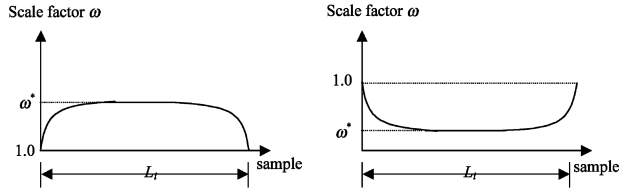
with i representing the associated GOS. After that, the results of E_{i1}, E_{i2}, E_{i3} are sorted and subsequently renamed to $E_{min}, E_{mid}, E_{max}$ for the corresponding minimum, middle and maximum value of the calculated AOAA. The binary value of the watermark is calculated using these differences:

$$A = E_{max} - E_{mid} \tag{18}$$

$$B = E_{mid} - E_{min} \tag{19}$$

If $A > B$, the watermark bit is interpreted as a "1". Otherwise ($A < B$), it is a "0".

If the actual interpreted value bit inside the GOS does not match the watermark bit, that has to be embedded in that case, signal corrections have to be applied. In the case of embedding a "1", the value of E_{max} has to be increased while decreasing E_{mid} by the same amount δ with $\delta > 0$ with the result of fulfilling the condition $A > B$. In the case of



(a) upscaling case (b) downscaling case

Figure 13: Curves used in the progressive scaling scheme (taken from [21])

embedding a "0", E_{mid} has to be increased while decreasing E_{min} by the same amount δ with $\delta > 0$ so that the condition $A < B$ is fulfilled.

In the current state, this embedding method would produce signal discontinuities in the time domain of the watermarked signal, exemplary depicted in Figure 12, leading to percieveble "click" sounds for human listeners. To address this problem, a progressive scaling scheme is used. By applying the curve in Figure 13(a) in the upscaling case, alternatively the curve in Figure 13(b) in the downscaling case, to the waveform of each dedicated section of the current watermark, a continuous waveform is achieved.

For decoding the watermark, knowledge about the sequence starting point and the section lengths L_1, L_2, L_3, used in the embedding process, is required for the decoder. If both is known, the AOAA values $E'_{i1}, E'_{i2}, E'_{i3}$ are calculated and sorted by their results to $E'_{min}, E'_{mid}, E'_{max}$ and subsequently the difference values A' and B' are calculated in the same manner as in the encoding process described before. If also the bit length of the embedded watermark is known, a majority decision, made between all counted occurrences of the extracted binary states at the corresponding watermark bit position, can be used to decode the final watermark.

4.1 Robustness and signal suitability

Both described methods show robustness against different signal manipulation attempts. However, there are still minor differences.

By compression of the watermarked signal with MP3 at a bit rate of 80 kb/s, the watermark detection performance in Bassia et al. still stays at 100% [3], while the recovery rate in Lie and Chang decreases to 97% [21]. Note that the detection performance in Bassia et al. also decreases for bit rates lower than 80 kb/s. While time scale modifications of the watermarked signal in Bassia et al. [3] lead to complete undetectability of the watermark, it is still possible in Lie and Chang for a time scaling of -3.1996%. [21]. However, the technique by Bassia et al. is robust against signal cropping, whereas this could destroy the watermarks in Lie and Chang's algorithm. The embedding bit rates of both algorithms are not mentioned by the authors.

Following the characterization of three distinct audio watermarking categories, a selection of current research topics in audio watermarking is discussed in the succeeding section.

5. CURRENT RESEARCH

Previously proposed audio watermarking techniques have still been a subject in current research. Based on the number of publications, the topic of echo watermarking has gained in importance in the last four years.

Xiang et al. presented a dual-channel time spread echo method for additional robustness [34]. It is based on the previously presented time spread echo method by Ko et al. [18] and is enhanced with two time spread echo kernels. In contrast to [18], the PN sequences used for the echo kernels can contain any real value and not merely the two values -1 and +1. In addition, the applied PN sequence can inhere the characteristics of colored noise and not merely white noise.

For embedding the watermark in the dual-channel time spread echo method [34], the original audio signal $x(n)$ is partitioned into two data sequences, each containing only even ($x_{even}(n)$) or odd ($x_{odd}(n)$) audio samples:

$$x_{even}(n) = [x(0), x(2), x(4), ...] \tag{20}$$

$$x_{odd}(n) = [x(1), x(3), x(5), ...] \tag{21}$$

With $q(n)$ describing the colored PN-sequence, α as the amplitude scaling factor and n_d as the sample distance between the first impulse $\delta(n)$ and the echo in the time domain, the echo kernels $h_1(n)$ and $h_2(n)$ are described as

$$h_1(n) = \delta(n) + \frac{\alpha}{2} * q(n - n_d) \tag{22}$$

and

$$h_2(n) = \delta(n) - \frac{\alpha}{2} * q(n - n_d) \tag{23}$$

With the convolution of $h_1(n)$ with $x_{even}(n)$ and $h_2(n)$ with $x_{odd}(n)$, the signals $y_{even}(n)$ and $y_{odd}(n)$ are generated. The final watermarked signal $y(n)$ is then created with the recombination of $y_{even}(n)$ and $y_{odd}(n)$ to the identical order as it was before the separation process (see Equations 20, 21):

$$y(n) = [y_{even}(0), y_{odd}(0), y_{even}(1), y_{odd}(1), ...] \tag{24}$$

For decoding the watermark information out of the watermarked audio signal, $y(n)$ is again partitioned in the same order as in the encoding process:

$$y_1(n) = [y(0), y(2), y(4)] = y_{even}(n) \tag{25}$$

$$y_2(n) = [y(1), y(3), y(5)] = y_{odd}(n) \tag{26}$$

By building the cepstrums of $y_1(n)$ of $y_2(n)$, named $c_{y1}(n)$ and $c_{y1}(n)$, the decoding functions for both signals result in

$$d_{y1}(\tau) = E(c_{y1}(n)q(n - \tau)) \tag{27}$$

$$d_{y2}(\tau) = E(c_{y2}(n)q(n - \tau)) \tag{28}$$

with E denoting the mathematical expectation value and τ the time value inside the cepstrum. Because the expectation values are sufficiently small, they can be rewritten to

$$d_{y1}(\tau) \approx E(c_{y1}(n)q(n - \tau)) + \frac{\alpha}{4} * r_{qq}(\tau) \tag{29}$$

$$d_{y2}(\tau) \approx E(c_{y2}(n)q(n - \tau)) - \frac{\alpha}{4} * r_{qq}(\tau) \tag{30}$$

with

$$r_{qq}(\tau) = E(q(n - n_d) * q(n - \tau)) \tag{31}$$

With the combination of both decoding functions, the final decoding function results in

$$\begin{aligned} d_{new}(\tau) &= d_{y1}(\tau) - d_{y2}(\tau) \\ &\approx E(r_x(n)q(n - \tau)) + \frac{\alpha}{4} * r_{qq}(\tau) \end{aligned} \tag{32}$$

with

$$r_x(n) = c_{x_{even}}(n) - c_{x_{odd}}(n) \tag{33}$$

which is the difference between the cepstrums of X_{even} and X_{odd}, known from the decoding process. If $d_{new}(n_0) > d_{new}(n_1)$, then the watermark bit is a "0" and vice versa it is a "1", which is similar to the other echo hiding methods described before.

Compared to the previously described time spread echo method by Ko et al. [18], the proposed dual channel time spread echo method showed overall higher robustness against signal manipulation attacks [34]. Nevertheless, both methods are less robust against pitch scaling [34].

Another contemporary research work on the time spread echo method was presented by Passi and Parmar [26], introducing three distinct types of PN sequences for the echo kernel. The sequences distinguish oneself in the binary sequence generation process. According to a listening test, conducted with three female and two male persons with *"normal listening capability"* [26], the audio signals watermarked with the three presented PN sequences lead to higher imperceptibility of the time spread echo.

Besides echo watermarking, Bliem et al. proposed a spread spectrum audio watermarking method robust for the situation of recording the watermarked audio signal, which is played back by a loudspeaker in room, with a microphone [5]. In contrast to previously described spread spectrum watermarking methods, microphone self noise is especially taken into account in the decoding process, which is why the signal amplitudes in the time domain are normalized approximately to the same level. Moreover, similar to the basic principle of the direct sequence spread spectrum technique, described in section 2.1, binary phase shift keying is applied to decode the watermark bit sequence.

For evaluating the robustness of the watermarking method in the given scenario, various objective performance tests were conducted, including simulations of reverberation and microphone movement. The reverberation test was simulated with the use of room impulse responses measured with microphones, placed at different positions in a semicircle around a loudspeaker, with the first located in front and the last positioned behind the loudspeaker [5]. The measured room impulse responses were convolved with the watermarked audio signal and afterwards, *"measured microphone and room noise of different levels"* was added to the resulting signal [5].

In the watermark detection process, the resulting signal exhibited an overall increase of the bit error rate, compared to the watermarked signal without reverberation [5]. Nevertheless, at a minimum signal-to-noise ratio of -19dB, almost

all watermark signals can be recovered [5]. The microphone movement was simulated with the usage of the Doppler effect, which is a frequency shift caused by a moving sound source, related to the listener [7]. The simulated microphone movement is defined as one-dimensional in the direction of the loudspeaker. The results show, that up to a velocity of 0.8 m/s, watermark signal decoding errors can be compensated in a *"real world scenario"*, according to the authors [5].

The exhibited excerpt of current research topics in audio watermarking shows, that longstanding watermarking methods still keep relevancy to date.

6. CONCLUSION

In this paper, three different techniques of audio watermarking have been presented and also the robustness properties and signal suitabilities of the presented approaches have been exhibited. With the exception of single echo hiding, all presented watermarking methods seem to be applicable to a wide range of audio signals, from speech to music signals. Though all of the presented techniques have different strengths and weaknesses in terms of robustness, most presented algorithms show, likewise, high robustness against lossy audio compression with MP3 or AAC at low audio encoding bitrates.

The highest challenge for all presented watermarking techniques is the robustness against time-scale modifications, resulting in synchronization errors and therefore decoding errors, which is why several counter-measures have been developed to date. A removal of the watermark might also be possible in very robust watermarking algorithms, but it will result in massive quality degradation (e.g. in [16]).

Since the robustness of a watermarked signal is dependent on the embedded amplitude of the watermark, a trade-off between robustness and imperceptibility exists [4]. By embedding the watermark with a high amplitude, the watermark might be very robust, but there might also be the possibility for the listener of perceiving the embedded watermark as signal artifacts. Since the auditory perception and evaluation of audio signal components is subjective and therefore dependent on every individual listener [25], there might also be listeners perceiving watermarked signals as annoying.

It is understandable, that the main priorities for the music industry are more in protecting their works by keeping the watermark robustness to a high degree, than in improving the perceived audio quality of watermarked audio signals. Nonetheless the listeners' demands on audio quality should always be taken seriously and therefore intended purpose of watermarking should always be pondered. For example in the case of digital distribution of high quality lossless music, a scenario aiming at music enthusiasts with high listening demands, perceivable watermarking artifacts might be disturbing to the target group and therefore lead to negative feedback from the listeners' side, as we could see in the real world example described in the introduction of this paper. So in this case, it should be considered carefully, whether audio watermarking would make sense in this scenario.

Since there are not many scientific listening tests known in literature, committed for the perception of watermarking artifacts, this might be subject for further research in audio watermarking. Especially with rising demands on audio quality from the listeners' side, this might be an important topic for the future.

7. REFERENCES

[1] M. Arnold. Audio watermarking: features, applications and algorithms. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 2, pages 1013–1016 vol.2, 2000.

[2] B. S. Atal and S. L. Hanauer. Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. *Journal of the Acoustical Society of America*, 50(2B):637–655, 1971.

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

[4] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM Systems Journal*, 35(3.4):313–336, 1996.

[5] T. Bliem, G. D. Galdo, J. Borsum, A. Craciun, and R. Zitzmann. A robust audio watermarking system for acoustic channels. *J. Audio Eng. Soc*, 61(11):878–888, 2013.

[6] L. Boney, A. Tewfik, and K. Hamdy. Digital watermarks for audio signals. In *Multimedia Computing and Systems, 1996., Proceedings of the Third IEEE International Conference on*, pages 473–480, Jun 1996.

[7] J. M. Chowning. The simulation of moving sound sources. *Computer Music Journal*, 1(3):48–52, 1977.

[8] I. J. Cox, J. Kilian, F. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. Technical report 95-10, NEC Research Institute, Princeton, NJ, USA, 1995.

[9] N. Cvejic, A. Keskinarkaus, and T. Seppanen. Audio watermarking using m-sequences and temporal masking. In *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*, pages 227–230, 2001.

[10] D. Gruhl, A. Lu, and W. Bender. Echo hiding. In R. J. Anderson, editor, *Information Hiding, First International Workshop, Cambridge, U.K., May 30 - June 1, 1996, Proceedings*, volume 1174 of *Lecture Notes in Computer Science*, pages 293–315. Springer, 1996.

[11] F. Hartung and M. Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, Jul 1999.

[12] X. Kang, R. Yang, and J. Huang. Geometric invariant audio watermarking based on an lcm feature. *Multimedia, IEEE Transactions on*, 13(2):181–190, April 2011.

[13] R. Kemerait and D. Childers. Signal detection and extraction by cepstrum techniques. *Information Theory, IEEE Transactions on*, 18(6):745–759, Nov 1972.

[14] D. Kirovski and H. Malvar. Robust spread-spectrum audio watermarking. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 3, pages

1345–1348 vol.3, 2001.

[15] D. Kirovski and H. Malvar. Spread-spectrum audio watermarking: requirements, applications, and limitations. In *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pages 219–224, 2001.

[16] D. Kirovski and H. Malvar. Spread-spectrum watermarking of audio signals. *Signal Processing, IEEE Transactions on*, 51(4):1020–1033, Apr 2003.

[17] B.-S. Ko, R. Nishimura, and Y. Suzuki. Time-spread echo method for digital audio watermarking using pn sequences. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 2, pages II–2001–II–2004, May 2002.

[18] B.-S. Ko, R. Nishimura, and Y. Suzuki. Time-spread echo method for digital audio watermarking. *Multimedia, IEEE Transactions on*, 7(2):212–221, April 2005.

[19] S.-K. Lee and Y.-S. Ho. Digital audio watermarking in the cepstrum domain. *Consumer Electronics, IEEE Transactions on*, 46(3):744–750, Aug 2000.

[20] W.-N. Lie and L.-C. Chang. Robust and high-quality time-domain audio watermarking subject to psychoacoustic masking. In *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, volume 2, pages 45–48 vol. 2, May 2001.

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

[22] Y. Lin and W. H. Abdulla. *Audio Watermark*. Springer, 2015.

[23] H. O. Oh, J. W. Seok, J. W. Hong, and D. H. Youn. New echo embedding technique for robust and imperceptible audio watermarking. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 3, pages 1341–1344 vol.3, 2001.

[24] A. Oppenheim and R. Schafer. From frequency to quefrency: a history of the cepstrum. *Signal Processing Magazine, IEEE*, 21(5):95–106, Sept 2004.

[25] T. Painter and A. Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515, April 2000.

[26] V. Passi and G. Parmar. Utilizing different types of sequences for audio watermarking based on time spread echo method. In *Industrial Instrumentation and Control (ICIC), 2015 International Conference on*, pages 1398–1401, May 2015.

[27] R. Pickholtz, D. Schilling, and L. Milstein. Theory of spread-spectrum communications–a tutorial. *Communications, IEEE Transactions on*, 30(5):855–884, May 1982.

[28] J. W. Seok and J. W. Hong. Audio watermarking for copyright protection of digital audio data. *Electronics Letters*, 37(1):60–61, Jan 2001.

[29] M. Steinebach, F. A. Petitcolas, F. Raynal, J. Dittmann, C. Fontaine, S. Seibel, N. Fates, and L. Ferri. Stirmark benchmark: audio watermarking attacks. In *Information Technology: Coding and Computing, 2001. Proceedings. International Conference on*, pages 49–54, Apr 2001.

[30] M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney. Robust audio watermarking using perceptual masking. *Signal Processing*, 66(3):337 – 355, 1998.

[31] R. Van Schyndel, A. Tirkel, and C. Osborne. A digital watermark. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 2, pages 86–90 vol.2, Nov 1994.

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

[33] S. Xiang and J. Huang. Histogram-based audio watermarking against time-scale modification and cropping attacks. *Multimedia, IEEE Transactions on*, 9(7):1357–1372, Nov 2007.

[34] Y. Xiang, I. Natgunanathan, D. Peng, W. Zhou, and S. Yu. A dual-channel time-spread echo method for audio watermarking. *Information Forensics and Security, IEEE Transactions on*, 7(2):383–392, April 2012.

[35] Y. Xiang, D. Peng, I. Natgunanathan, and W. Zhou. Effective pseudonoise sequence and decoding function for imperceptibility and robustness enhancement in time-spread echo-based audio watermarking. *Multimedia, IEEE Transactions on*, 13(1):2–13, Feb 2011.

[36] C. Xu, J. Wu, Q. Sun, and K. Xin. Applications of digital watermarking technology in audio signals. *J. Audio Eng. Soc*, 47(10):805–812, 1999.

[37] I.-K. Yeo and H. J. Kim. Modified patchwork algorithm: a novel audio watermarking scheme. In *Information Technology: Coding and Computing, 2001. Proceedings. International Conference on*, pages 237–242, Apr 2001.