Robust-audio-hash Synchronized Audio Watermarking

Martin Steinebach, Sascha Zmdzinski, Sergey Neichtadt

Fraunhofer IPSI, Dolivostrasse 15,
64293 Darmstadt, Germany

Abstract. Digital audio watermarking has become an accepted technology for e.g. protection of music downloads. While common challenges to robustness, like lossy compression or analogue transmission have been solved in the past, loss of synchronization due to time stretching is still an issue. We present a novel approach to audio watermarking synchronization where a robust audio hash is applied to identify watermarking positions.

1 Motivation

Digital watermarking is a technique to embed hidden information imperceptibly into multimedia data. Watermarking schemes consist of an embedding stage and a retrieving stage: In the embedding stage the hidden information is embedded using a secret watermark key into the cover file. In the retrieving stage the watermark can be detected and retrieved given that the secret key is known at retrieving time. The most important requirements for digital watermarking are known to be [5] transparency, robustness, capacity, security and complexity. Algorithms are known primarily for image and audio data, but various other media and data formats are also addressed by watermarking. In the following work we address audio watermarking.

One well-known weakness of watermarking algorithm are de-synchronization attacks [9]. Here the embedded watermark information is not removed from a media file but only slightly moved to a position where the watermark detection algorithm will not try to retrieve the watermark. Time stretching, the slight increase or decrease of audio playing time without pitch modification or significant quality loss, is one known example for de-synchronization attacks on audio material.

Figure 1 shows how de-synchronization by time stretching works: On the left side a typical example of an audio watermark is given. Individual watermarking bits #1 to #3 are embedded sequentially in frames of a defined length. A detection algorithm will synchronize with the help of a sync signal then will try to retrieve #1 to #3. On the right side the effects of a time-stretching attack are shown. The audio material and the embedded watermarking frame are now longer than before. After synchronization the retrieval process will detect #1 and also may be able to correctly retrieve #2. But the original frame length will make the retriever try to detect #3 at a position where now still #2 is in effect as the frames are now longer. Retrieval errors are the consequence.

Steinebach M., Zmdzinski S. and Neichtadt S. (2006).
Robust-audio-hash Synchronized Audio Watermarking.
In *Proceedings of the 4th International Workshop on Security in Information Systems*, pages 58-66

This is obviously a challenge where a repetitive re-synchronization would help. But synchronization in audio watermarking often requires much of the watermarking capacity. Therefore synchronization after each bit would render an algorithm robust but useless due to minimal capacity.

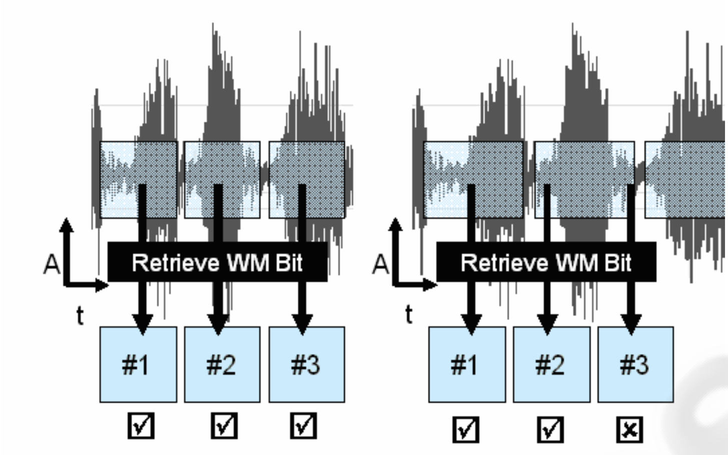

Fig. 1. Effect of time stretching on watermark retrieval. Left: before time stretching, right: after time stretching.

We propose an alternative solution to this challenge. Our proposed algorithm does not require embedded sync sequences to synchronize the watermarking bits but uses robust audio hashing technology to re-synch at each embedded bit.

2 Background

In this section we briefly describe the two basic technologies applied in our novel approach, digital watermarking and robust audio hashing. Both are combined in section 3 to a new watermarking concept.

2.1 Digital Audio Watermarking

Digital watermarking schemes have been under research and development for various types of multimedia data for many years, including audio formats like PCM, mp3 or MIDI. In this work we focus on digital PCM audio data. Several approaches for PCM audio watermarking have been introduced in the literature, like in [2], in [5] or [10].

The latter algorithm is the base of the watermarking part of our new approach. It embeds an information bit into the frequency representation of a frame of 2048 samples. The resulting frequency bands are pseudo-randomly selected and associated to two groups A and B. The value of the information bit is defined by the difference of energy levels of A and B. If A > B means “0”, B > A equals “1”. Watermark embedding is done by enforcing these energy differences by modifying the frequency bands of A and B under the control of a psychoacoustic model.

2.2 Robust Audio Hash

Robust audio hash algorithms have also been called audio IDs or audio fingerprints in the literature. The concept here is to derive a robust content-dependent description from audio data to able to identify the audio data by comparing the stored and a newly calculated description. This description is much more robust to modifications of the audio data, like e.g. mp3 compression, than a cryptographic hash would be. Various approaches for deriving a robust content description have been introduced [4], [1].

In this work we adapt the algorithm introduced in [7] where the robust hash is based on the relation of energy levels of frequency bands. A robust hash of 31 bits is calculated by comparing the energy of a frequency band to its predecessor in time and its lower neighbor in the spectrum.

Other known concepts [8] include the inherent periodicity of audio signals, the time-frequency landscape given by the frame-by-frame mel-frequency cepstral coefficients, principal component analysis, adaptive quantization and channel decoding.

2.3 Robust Hash Algorithms and Digital Watermarking

In the literature first approaches to combine robust hashing and digital watermarking have been discussed. In the video domain, in [6] the authors use robust hashes extracted at the watermark embedding position and stored in a database to later re-synchronize the watermark. The marked video is scanned for the hash stored in the database and the watermark is retrieved at the position the hash is found. For audio, in [3] a method also is proposed to use extracted and stored hashes to re-synchronize the watermarks. While this method may help to retrieve the embedded watermarks, the obvious drawback the need to have the stored hashes available at watermark retrieval. This leads to a sort of semi-non-blind watermarking.

In the rest of this paper we will present an approach which also uses robust hashes to re-synchronize the embedded watermarking information, but does not require stored hash information.

3 Algorithm Design

In this section we introduce our novel audio watermarking concept combining audio watermarking and robust audio hashing.

3.1 Concept

Our approach uses digital watermarking to embed information bits into audio data. A robust audio hash is applied for synchronization, thereby circumventing the common need for audio watermarking synchronization. Figure 2 illustrates this process:

1. First the audio hash is retrieved from a small frame of the audio file.
2. Then we check if the hash is linked to a watermarking bit
3. If the hash is not linked to a bit, the frame position is increased an the algorithm starts again at (1)
4. If the hash is linked to a bit, the number of the watermarking bit is identified
5. The watermarking algorithm is used to embed the watermarking bit at the position of the retrieved hash
6. The frame position is increased and the algorithm starts at (1)

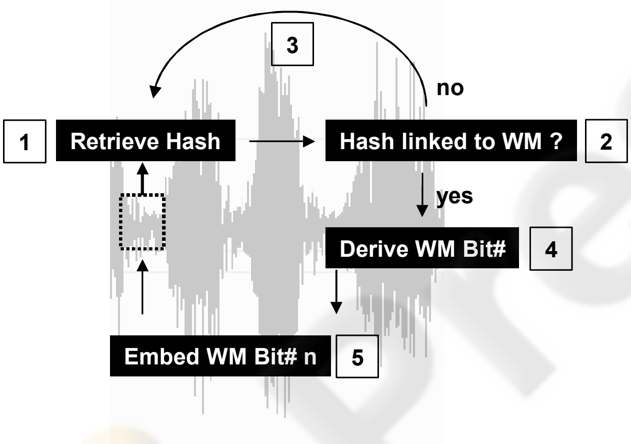

Fig. 2. The general concept is to retrieve a hash, check if the hash is assigned to a watermarking bit, identify the bit number and embed the watermarking bit at the hash position. The numbers identify the step described in the text above.

For watermarking retrieval, the process is:

1. The marked file is scanned for hashes linked to a watermarking bit
2. If the hash is linked to a bit, the number of the watermarking bit is identified
3. The watermarking algorithm is used to retrieve the watermarking bit at the position of the detected hash
4. After the whole audio file is scanned, the complete watermark is made available by putting the distributed and retrieved watermarking bits back in order with the help of the hash indices.

To enable this algorithm, we need a set of audio hashes which are linked to a watermarking bit. We need at least as many hashes as there are watermarking bits to be embedded, but allocation of more than one hash to a single watermarking bit is possible. To ensure not each frame position is used for embedded which would cause an overlay of watermarking information and thereby transparency and robustness problems, only a small amount of all possible audio hashes should be assigned to watermarking bits (see figure 3). The rest of the hashes are ignored when detected.

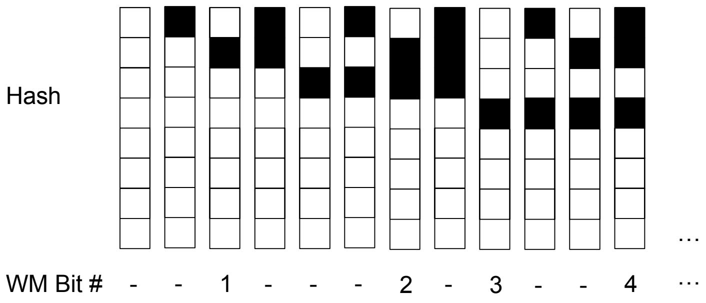

Fig. 3. Hash allocation. Note that not every hash is linked to a watermarking bit. Starting from the left, the third hash is assigned to watermarking bit #1.

Using the hashes in this manner enables to scan through an audio file while watermark embedding or retrieval for position where one bit of the complete watermark has been embedded or has to be embedded. It therefore works as a sort of index to the audio file, not only pointing to embedding positions, but also providing the information which bit of a watermark is allocated to the current audio frame in the case the hash is allocated to a watermarking bit. Figure 4 illustrates this.

3.2 Robust Audio Hash Optimization

The robust hash algorithm introduced in [7] has been designed to distinguish between large numbers of musical pieces. It is therefore rather sensitive to small differences within the audio content. This makes it hard to use for watermark synchronization as attacks on the watermark would change the audio hashes. The result would either be a wrong watermarking bit allocation or no detected bit at all.

Therefore we reduced the hash resolution from 32 to 6 bits to increase the hash robustness. The number of possible different hash values is now 64, so in theory the maximum watermarking payload would be 64 bit. But as not all positions must be used to prevent overlapping while embedding and retrieval, our maximum payload is 42 bits. The rest of the hash values are not used and act as gaps in the embedding process.

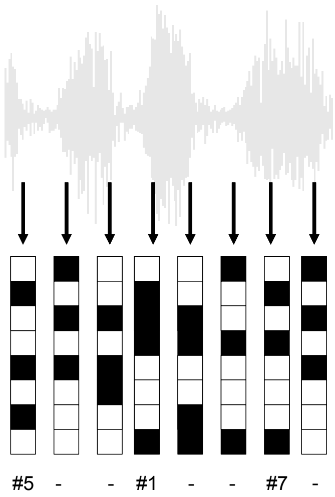

Fig. 4. Retrieved audio hashes are assigned to watermarking bits.

The selection of the hashes not used for watermark synchronization has been done on a statistical analysis of hash occurrences in audio files. These are not equally distributed, but some hashes tend to appear very often while others are rather rare. Figure 5 shows the distribution of hashes within on audio piece. We used a large number of audio files to identify those hashes which tend to occur regularly but not extremely often and used these for watermark embedding. The hashes which occurred extremely often or very seldom were used as gap hashes to ensure an equal distribution of embedded watermarking bits.

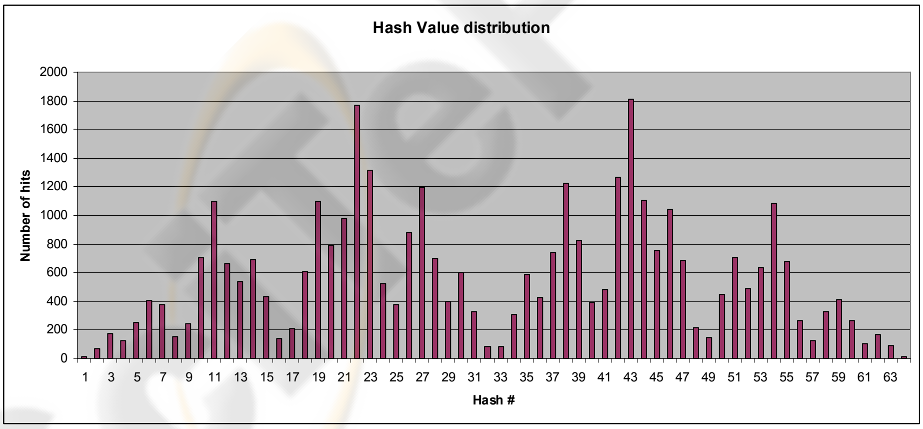

Fig. 5. Hash distribution is not equal in audio material.

The algorithm will only be as robust and reliable as the basic robust audio hash function. We therefore did an intense analysis of the error rates of our robust audio hash version with its reduced number of hash bits. A selection of results is shown in figure

6. Bit error rates for the 6 bit hashes are given for four audio files and attacks from equalization to mp3 compression. As only a hash which is derived correctly from the audio data addresses the right watermarking bit, one bit error in the hash can be seen as a complete failure of the retrieval process at this position. As the watermark bit will be embedded several times within the audio file, the hash bit errors not automatically lead to a failure at watermark retrieval. Error correction is applied to ensure a high robustness versus singular errors in the retrieved bit sequences and the values of the individual bit positions are chosen by calculating the strongest watermarking bit signal over all positions assigned to this bit number by the hashes.

The bit error rates in figure 6 are at their highest after mp3 compression with 96 kbps. Here the bit error rate is about 12% for a white noise signal. The average error rate is at 5%. This is sufficiently low to be removed by selection of the strongest signal and error correction. The 6 bit robust audio hash is therefore robust against attacks assumed to be relevant for audio watermarking. The overall error rate of the hash-based audio watermarking system is a sum of hash errors and watermarking retrieval errors.

Table 1. Robustness of our watermarking algorithm against (A) mp3 vbr 128 kbps, (B) mp3 vbr 88 kbps, (C) Time Strech 2%, (D) Time Strech 3%, (E) Pitch Shift 3%, (F) Time Shift 15 Samples, (G) DA/AD conversion, (H) cropping. BE stands for "bit error" and shows how many wrong watermarking bits have been retrieved.

Audio File	Attack	BE
Pop (1 min)	A	0
Rock1 (song)	A	0
Rock2 (song)	A	0
Classic (1 min)	A	0
Classic (1 min)	none	0
Classic (song)	A	0
Rock1 (1 min)	A	2
Rock2 (55 s)	A	0
Alternative(1 min)	A	0
Alternative(1 min)	none	0
Alternative(1 min)	A + H	0
Alternative(1 min)	B	20
Alternative(song)	B	4
Alternative(1 min)	C	1
Alternative(song)	D	0
Alternative(1 min)	E	43
Alternative(1 min)	F	3
Alternative(song)	F	0
Alternative(song)	G	2

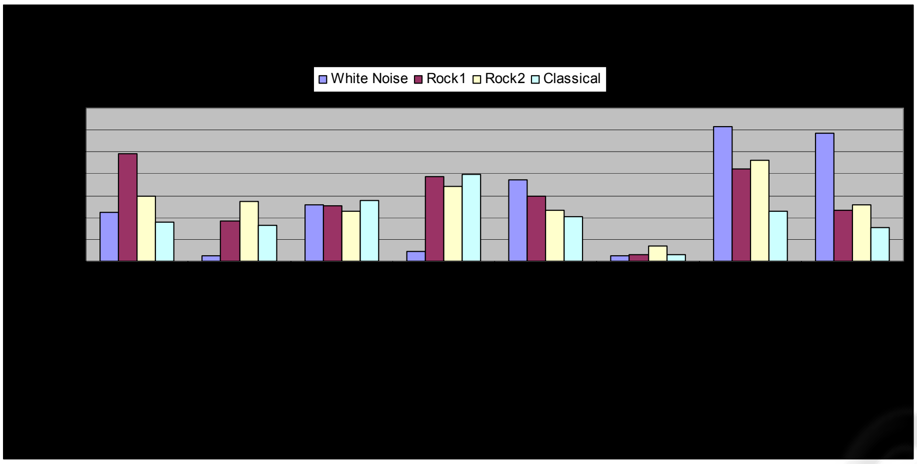

Fig. 6. Robust audio hash robustness versus various attacks.

4 Summary and Future Work

The focus of this paper is the presentation of a novel audio watermarking approach where robust audio hashing is used for synchronization without the need to store the audio hash data. We use a selected set of hashes derived from a large number of audio files and use these as a general index to audio data. The hashes are assigned to single bits of a watermark message. They are used to identify the position to embed and later to retrieve the bit.

First test results show the good robustness against time stretching, an attack which is often a challenge for watermark embedding. Other test results show the characteristics of the hash distribution within audio material. While the hashes are not evenly distributed, a subset of hashes with such a characteristic can be found.

For future work we need to improve possible watermark payload and watermark robustness. To achieve this, we are currently working on a two-layer approach, where one hash is used to identify an embedding or retrieval position and a second hash at this position is used to assign the corresponding watermark bit. This has a positive effect on the algorithm: The first hash is chosen in a way the distance between two embedding positions will be acceptable in most cases. The second has will be distributed in a way all watermarking bits will be embedded a similar number of times within the audio file.

Acknowledgements

The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the authors' views, is provided as is and no

guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

1. Allamanche, Herre, Helmuth, Fröba, Kasten, Cremer; Content-Based Identification of Audio Material Using MPEG-7 Low Level Description, in electronic Proceedings of the International Symposium of Music Information Retrieval, http://ismir2001.ismir.net/papers.html, 2001
2. Laurence Boney, Ahmed H. Tewfik and Khaled N. Hamdy, Digital Watermarks for Audio Signals, 1996 IEEE Int. Conf. on Multimedia Computing and Systems June 17-23, Hiroshima, Japan, p. 473-480.
3. Beauget, S., van der Veen, M., and Lemma, A. 2004. Informed detection of audio watermark for resolving playback speed modifications. In Proceedings of the 2004 Workshop on Multimedia and Security (Magdeburg, Germany, September 20 - 21, 2004). MM&Sec '04. ACM Press, New York, NY, 2004
4. P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of algorithms for audio fingerprinting. In International Workshop on Multimedia Signal Processing, US Virgin Islands, December 2002
5. Cox, Miller, Bloom; Digital Watermarking, Academic Press, San Diego, USA, ISBN 1-55860-714-5, 2002
6. O. Harmanci, M. Kucukgoz, M. Mihcak: Temporal synchronization of watermarked video using image hashing, In Proc of IEEE Security, Steganography and Watermarking of Multimedia Contents VII, Volume 5681, San Jose, USA, pp. 370-380, January 2005
7. J. Haitsma, T. Kalker, and J. Oostveen, "Robust Audio Hashing for Content Identification," in Proceedings of the Content-Based Multimedia Indexing, 2001
8. Özer, Sankur, Memon, Robust Audio Hashing for Audio Identification, EUSPICO, 2005
9. Steinebach, Martin; Petitcolas, Fabien A. P.; Raynal, Frederic; Dittmann, Jana; Fontaine, Caroline; Seibel, Christian; Fates, Nazim; Croce Ferri, Lucilla (2001). StirMark Benchmark: Audio watermarking attacks. In: Int. Conference on Information Technology: Coding and Computing (ITCC 2001), April 2 - 4, Las Vegas, Nevada, pp. 49 - 54, ISBN 0-7695-1062-0, 2001.
10. Steinebach, Digitale Wasserzeichen für Audiowasserzeichen, ISBN 3-8322-2507-2, Shaker Verlag, 2004