EURASIP Journal on Applied Signal Processing 2003:10, 1001–1015
© 2003 Hindawi Publishing Corporation

Watermarking-Based Digital Audio Data Authentication

Martin Steinebach
Fraunhofer Institute IPSI, MERIT, C4M Competence for Media Security, D-64293 Darmstadt, Germany
Email: martin.steinebach@ipsi.fraunhofer.de

Jana Dittmann
Platanista GmbH and Otto-von-Guericke University Magdeburg, 39106 Magdeburg, Germany
Email: jana.dittmann@iti.cs.uni-magdeburg.de

Received 11 July 2002 and in revised form 4 January 2003

Digital watermarking has become an accepted technology for enabling multimedia protection schemes. While most efforts concentrate on user authentication, recently interest in data authentication to ensure data integrity has been increasing. Existing concepts address mainly image data. Depending on the necessary security level and the sensitivity to detect changes in the media, we differentiate between fragile, semifragile, and content-fragile watermarking approaches for media authentication. Furthermore, invertible watermarking schemes exist while each bit change can be recognized by the watermark which can be extracted and the original data can be reproduced for high-security applications. Later approaches can be extended with cryptographic approaches like digital signatures. As we see from the literature, only few audio approaches exist and the audio domain requires additional strategies for time flow protection and resynchronization. To allow different security levels, we have to identify relevant audio features that can be used to determine content manipulations. Furthermore, in the field of invertible schemes, there are a bunch of publications for image and video data but no approaches for digital audio to ensure data authentication for high-security applications. In this paper, we introduce and evaluate two watermarking algorithms for digital audio data, addressing content integrity protection. In our first approach, we discuss possible features for a content-fragile watermarking scheme to allow several postproduction modifications. The second approach is designed for high-security applications to detect each bit change and reconstruct the original audio by introducing an invertible audio watermarking concept. Based on the invertible audio scheme, we combine digital signature schemes and digital watermarking to provide a public verifiable data authentication and a reproduction of the original, protected with a secret key.

Keywords and phrases: multimedia security, manipulation recognition, content-fragile watermarking, invertible watermarking, digital signature, original protection.

1. INTRODUCTION

Multimedia data manipulation has become more and more simple and undetectable by the human audible and visual system due to technology advances in recent years. While this enables numerous new applications and generally makes it convenient to work with image, audio, or video data, a certain loss of trust in media data can be observed. As we see in Figure 1, small changes in the audio stream can cause a different meaning of the whole sentence.

Regarding security particularly in the field of multimedia, the requirements on security increase. The possibility and the way of applying security mechanisms to multimedia data and their applications need to be analyzed for each purpose separately. This is mainly due to the structure and complexity of multimedia, see, for example, [1]. The security requirements such as integrity (unauthorized modification of data) or data authentication (detection of origin and data alterations) can be met by the succeeding security measures using cryptographic mechanisms and digital watermarking techniques [1]. Digital watermarking techniques based on steganographic systems embed information directly into the media data. Besides cryptographic mechanisms, watermarking represents an efficient technology to ensure both data integrity and data origin authenticity. Copyright, customer, or integrity information can be embedded, using a secret key, into the media data as transparent patterns. Based on the application areas for digital watermarking known today, the following five watermarking classes are defined: authentication watermarks, fingerprint watermarks, copy control watermarks, annotation watermarks, and integrity watermarks. The most important

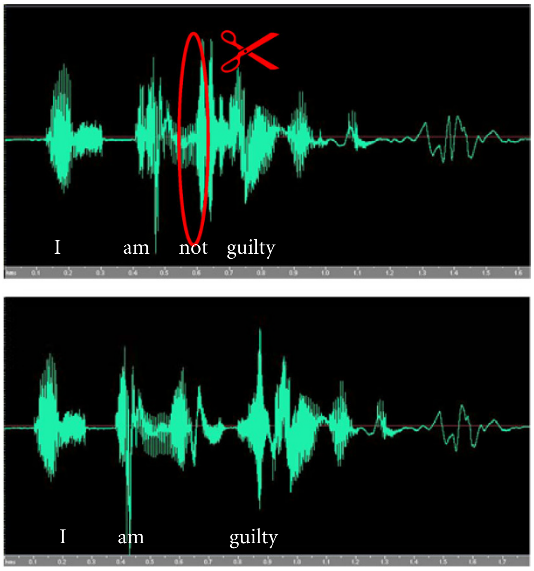

Figure 1: Digital audio data is easily manipulated.

properties of digital watermarking techniques are robustness, security, imperceptibility/transparency, complexity, capacity, and possibility of verification and invertibility, see, for example, [2].

Robustness describes whether the watermark can be reliably detected after media operations. It is important to note that robustness does not include attacks on the embedding schemes that are based on the knowledge of the embedding algorithm or on the availability of the detector function. Robustness means resistance to "blind," nontargeted modifications, or common media operations. For example, the Stirmark tool [3] attacks the robustness of watermarking algorithms with geometrical distortions. For manipulation recognition, the watermark has to be fragile to detect altered media.

Security describes whether the embedded watermarking information cannot be removed beyond reliable detection by targeted attacks based on full knowledge of the embedding and detection algorithm and possession of at least one watermarked data. Only the applied secret key remains unknown to the attacker. The concept of security includes procedural attacks or attacks based on a partial knowledge of the carrier modifications due to message embedding. The security aspect also includes the false positive detection rates.

Transparency relates to the properties of the human sensory system. A transparent watermark causes no perceptible artifacts or quality loss.

Complexity describes the effort and time we need to embed and retrieve a watermark. This parameter is essential for real-time applications. Another aspect addresses whether the original data is required in the retrieval process or not. We distinguish between nonblind and blind watermarking schemes, the latter require no original copy for detection.

Capacity describes how many information bits can be embedded into the cover data. It also addresses the possibility of embedding multiple watermarks in one document in parallel.

The *verification* procedure distinguishes between private verification similar to symmetric cryptography and public verification like in asymmetric cryptography. Furthermore, during verification, we differ between *invertible and noninvertible techniques*, where the first one allows the reproduction of the original and the last one provides no possibility to extract the watermark without alterations of the original.

The optimization of the parameters is mutually competitive and cannot be clearly done at the same time. If we want to embed a large message, we cannot require strong robustness simultaneously. A reasonable compromise is always a necessity. On the other hand, if robustness to strong distortions is an issue, the message that can be reliably hidden must not be too long.

Therefore, we find different kinds of optimized watermarking algorithms. The robust watermarking methods for owner and copyright holder or customer identification are usually unable to detect manipulations in the cover media and their design is completely different from that of fragile watermarks. When dealing with fragile watermarks, different aspects of manipulation have to be taken into account.

A fragile watermark is a mark that is easily altered or destroyed when the host data is modified through a linear or nonlinear transformation. The sensitivity of fragile watermarks to modification leads to their use in media authentication. Today we find several fragile watermarking techniques to recognize manipulations. For images, Lin and Delp [4] summarize the features of fragile schemes and their possible attacks. Fridrich [5] gives an overview of existing image techniques. In general, we can classify the techniques as ones which work directly in the spatial domain or in the transform (DCT, wavelet) domains. Furthermore, Fridrich classifies fragile (very sensitive to alterations), semifragile (less sensitive to alterations), visual-fragile (sensitive to visual alterations) watermarks (here we can generalize such schemes into content-fragile watermarks), and self-embedding watermarking as a means for detecting both malicious and inadvertent changes to digital imagery.

Altogether, we see that the watermarking community in favor of robust techniques has neglected fragile watermarking for audio data. There are only few approaches and many open research problems that need to be addressed in fragile watermarks, for example, the sensitivity to modifications [6]. The syntax (bit stream) of multimedia data can be manipulated without influencing their semantics, as it is the case with scaling, compression, or transmission errors. Thus it is more important to protect the semantics of the data instead of their syntax to vouch for their integrity. Therefore, content-based watermarks [7] can be used to verify illegal manipulations and to allow several content-preserving operations. Therefore, the main research challenge is to differentiate between content-preserving and content-changing manipulations. Most existing techniques use threshold-based techniques to decide the content integrity. The main problem is to face the wide variety of allowed content-preserving operations. As we see in the literature, most algorithms address the problem of compression. But very often, scaling, format conversion, or filtering are also allowed transformations.

Furthermore, for high-security application, we have the requirement to detect each bit change in an audio track and to extract the watermark embedded as additional noise. Invertible schemes face this problem and have been introduced for image and video data in recent publications [8]. To ensure a public verification, these approaches have been combined with digital signatures by Dittmann et al. [9]. As we see from the literature, there are no approaches for an invertible audio watermarking scheme.

Our contribution focuses mainly on the design of a content-fragile audio watermarking scheme to allow several postproduction processes and on the design of an invertible watermarking scheme combined with digital signatures for high-security applications. We introduce two watermarking algorithms: our first approach is a content-fragile watermarking scheme combining fragile feature extraction and robust audio watermarking, and the second approach is designed to detect each bit change and reconstruct the original audio, where we combine digital signature schemes and digital watermarking to provide a public verifiable data authentication and a reproduction of the original protected with a secret key.

In the following subsections, we firstly review the state of the art of basic concepts for audio data authentication; secondly, we describe the general approaches for content-fragile and invertible schemes as basis for our conceptual design in Sections 2 and 3. In Section 4, we show example applications, and we summarize our work in Section 5.

1.1. Digital audio watermarking parameters and general methods for data authentication

There are numerous algorithms for audio watermarking; as selection, see [10, 11, 12, 13, 14, 15, 16]. Most of them are designed as copyright protection mechanisms, and therefore, the robustness, security, capacity, and transparency are the most important design issues, while in a lot of approaches, complexity and possible verification methods come second.

In the case of fragile watermarking for data authentication, the importance of the parameters changes. The fragility and security with a moderate transparency are most important. Depending on what kind of fragility we expect, remember that we differentiate between fragile, semifragile, content-fragile, self-embedding, and invertible schemes; a high payload of the watermarking algorithm is necessary to embed sufficient data to verify the integrity. Security is important as the whole idea of fragile watermarking is to provide integrity security, and a weak watermarking security would mean a weak overall system as embedded information could be forged. Using cryptography while embedding, the data can further increase security, for example, asymmetric systems could be used to ensure the authenticity of the embedded content descriptions. Robustness is not as important as security. If, due to media manipulations, a certain loss of quality is reached and the content is changed or is not recognizable any more, the watermark can be destroyed. Depending on the application transparency can be less important as content protected by this scheme is usually not to be used for entertainment with high-end quality demands. Complexity can become relevant if the system is to work in real time, which is the case if it is applied directly into recording equipment like cameras.

Fragile watermarking can also be applied to audio data. If the algorithm is fragile against an attack, the watermark cannot be retrieved afterwards. Therefore, not being able to detect a watermark in a file, which is assumed to be marked, identifies a manipulation.

Content-fragile watermarks discriminate between content-preserving and content-manipulating operations. In the literature, we find only few approaches for audio authentication watermarks. In [17], the focus of audio content security has been on speech protection. Wu and Kuo describe two methods for speech content authentication. The first one is based on content feature extraction integrated in CELP speech coders. Here, content-relevant information is extracted, encrypted, and attached as header information. The second one embeds fragile watermarks in content-relevant frequency domains. They stress the fact that common hash functions are not suited for speech protection because of unavoidable but content-preserving addition of noise during transmission and format changes. Feature extraction and watermarking are both regarded as a more robust alternative to such hash functions. Wu and Kuo provide experimental results regarding false alarms and come to the conclusion that discrimination between weak content-preserving operations and content manipulations is possible with both methods. This is similar to our results provided in Section 2.

Dittmann et al. [18] introduce a content-fragile watermarking concept for multimedia data authentication, especially for a/v data. While previous data authentication watermarking schemes address a single media stream only, the paper discusses the requirements of multimedia protection techniques, where the authors introduce a new approach called 3D thumbnail cube. The main idea is based on a 3D hologram over continuing video and audio frames to verify the integrity of the a/v stream.

1.2. Feature-based authentication concept: content-fragile watermarking

As introduced, the concept of a content-fragile watermark combines a robust watermark and a content abstraction from a feature extraction function for integrity verification. During verification, the embedded content features are compared with the actual content, similar to hash functions in cryptography. If changes are detected, content and watermark differ, a warning message is prompted. The idea of content-fragile watermarking is based on the knowledge that we have to handle content-preserving operations, manipulations that do not manipulate the content.

Two different approaches of content embedding strategies can be recognized: direct embedding and seed-based embedding. With the first approach, a complete feature-based content description is embedded in the cover signal (original). The second approach uses the content description to generate information packages of smaller size based on the extracted features.

Direct embedding. In direct embedding, the extracted features are embedded bit by bit into the corresponding media data. The feature description has to be coded as a bit vector to be embedded in this way. The methods of embedding differ for every watermarking algorithm. What they have in common is that the feature vector is the embedded watermarking information. The problem with direct embedding is the payload of the watermarking technology: to embed a complete and sufficiently exact content description, very high bit rates would be necessary, which most watermarking algorithms cannot provide.

Seed-based approach. Features are used to achieve robustness against allowed media manipulations while still being able to detect content manipulations. The amount of data for the describing features is much less than the described media. But usually, even this reduced data cannot be embedded into the media as a watermark. The maximum payload of today's watermarking algorithms is still too small. Therefore, to embed some content description, we have to use summaries or very global features—like the root mean square (RMS) of one second of audio. This leads to security problems: if we only have information about a complete second, parts smaller than a second could be changed or removed without being noticed. A possible solution is to use a seed-based approach. Here, we use the extracted features as an addition to the embedding key. The embedding process of the watermark now depends on the secret key and the extracted features. The idea is that only if the features have not been changed, the watermark can be extracted correctly. If the features are changed, the retrieval process cannot be initialized to read the watermark.

In Section 2, we introduce a content-fragile audio watermarking algorithm based on the direct embedding strategy.

Remark 1. There are also more simple concepts of audio data authentication, which we do not address here, as they include no direct connection with the content. For example, embedding of a continuous time code is a way to recognize cutout attacks. The retrieved time code will show gaps at the corresponding positions if a sufficiently small step size has been chosen.

1.3. Invertible concept

The approach in [19] has introduced the first two invertible watermarking methods for digital image data. While virtually all previous authentication watermarking schemes introduced some small amount of noninvertible distortion in the data, the new methods are invertible in the sense that if the data is deemed authentic, the distortion due to authentication can be completely removed to obtain the original data. Their first technique is based on lossless compression of biased bit streams derived from the quantized JPEG coefficients. The second technique modifies the quantization matrix to enable lossless embedding of one bit per DCT coefficient. Both techniques are fast and can be used for general distortion-free (invertible) data embedding. The two methods provide new information assurance tools for integrity protection of sensitive imagery, such as medical images or high-importance military images viewed under nonstandard conditions when usual criteria for visibility do not apply. Further improvements in [8] generalize the scheme for compressed image and video data.

In [9], an invertible watermarking scheme is combined with a digital signature to provide a public verifiable integrity. Furthermore, the original data can only be reproduced with a secret key. The concept uses the general idea of selecting public key dependent watermarking positions (here, e.g., the blue channel bits) and compressing the original data at these positions losslessly to produce space for invertible watermark data embedding. In the retrieval, the watermarking positions are selected again, the watermark is retrieved, and the compressed part is decompressed and written back to recover the original data. The scheme is highly fragile and the original can only be reproduced if there was no change. The integrity of the whole data is ensured with two hash functions: the first is built over the remaining image and the second over the marked data at the watermarking positions by using a message authentication code HMAC. The authenticity is granted by the use of an RSA digital signature. The reproduction by authorized persons is granted by a symmetric key scheme: AES. The protocol for image data from [9] can be written as follows:

$$
\begin{aligned}
I_W &= I_{\text{remaining}} \| W \| \text{Data}_{\text{info}} // \text{Data}_{\text{fill}},\\
W &= E_{\text{AES}}\big(E_{\text{AES}}(\text{C}_{\text{blueBits}}, k_H(I_{\text{remaining}})), K_{\text{secret}}\big)\\
&\qquad //\text{HMAC}(\text{Selected}_{\text{blueBits}}, K_{\text{secret}})\\
&\qquad //\text{RSA}_{\text{signature}}\big(H\big(I_{\text{remaining}}\\
&\qquad //E_{\text{AES}}(E_{\text{AES}}(\text{C}_{\text{blueBits}}, k_H(I_{\text{remaining}}), K_{\text{secret}})\\
&\qquad //\text{HMAC}(\text{selected}_{\text{blueBits}}, K_{\text{secret}})\big), K_{\text{private}}\big).
\end{aligned}
$$

The watermarked image data I_W contains the remaining nonwatermarked image bits $I_{\text{remaining}}$ and the image data at the watermarking bit positions derived from the public key (see cursive in the equation) where the watermark is placed. The watermark data W itself contains the compressed original data C of the marking position bits, which are encrypted with the function E by AES using an encryption key $k_H(I_{\text{remaining}})$ derived by the hash value from the remaining image to verify the integrity. As invertibility protection [9], use an additional AES encryption E of the first encryption with the secret key parameter K_{secret} only known by authorized persons. To ensure the integrity of the original compressed data at the marking positions [9], use an HMAC function initialized by the secret key too. To enable public verification, the authors add an additional private key initialized *RSA* signature, which is built over the hash value of the remaining image, twice encrypted compressed data, and the HMAC function. For synchronization in the retrieval, information about the selected watermarking positions and the used compression function is added as well as padding bits. To verify the integrity and authenticity of the data, the user can use the public key to retrieve the watermark information

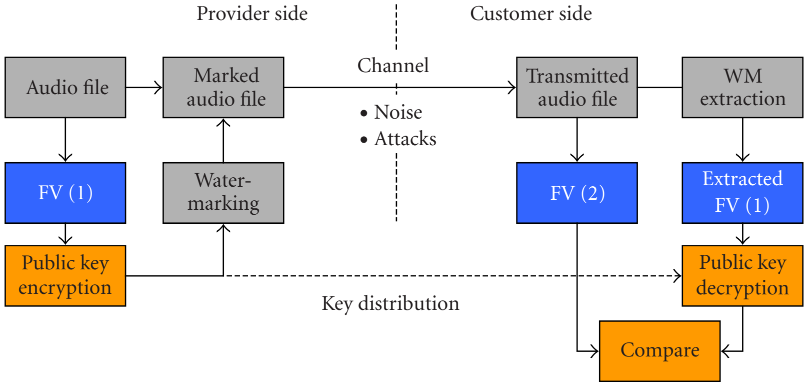

Figure 2: Content-fragile data authentication scheme.

and verify the RSA signature with the public key. For original reproduction, the secret k is necessary to decrypt the compressed data. With the HMAC function, the authenticity and integrity of the decrypted original data can be ensured. The general scheme can be described as dividing the digital document into two sets A and B. The set A is kept unchanged. The set B will be severed as a cover for watermark embedding, where B is compressed to C to produce room for embedding the digital signature S. To ensure that C belongs to A, we encrypt C with a content-depending key derived from A, and to restrict reproduction of original C, it is again encrypted with a secret key. The digital signature S is built over A and the twice encrypted C as well as the message authentication code to ensure correct reproduction of C.

In our paper, we adopt the scheme of [9] for digital audio data and introduce a new invertible audio watermark, see Section 3.

2. CONTENT-FRAGILE AUDIO WATERMARKING

In this section, we introduce our approach to content-fragile audio watermarking based on the concepts introduced in Section 1.2. We address suitable features of audio data, introduce an algorithm, and provide test results.

2.1. *Content-fragile authentication concept*

Figure 2 illustrates the general content-fragile audio watermarking concept: from an audio file, a feature vector (FV) is extracted and may be encrypted. This information is embedded as a watermark. The audio file is then transmitted via a noisy channel. At some time, the content has to be verified. Now the watermark (WM) is extracted and the embedded and decrypted FV is compared to a newly generated FV. If a certain level of difference is reached, integrity cannot be verified. A PKI may be helpful to handle key management.

Remember, fragility is about losing equality of extracted and embedded contents in this case with the challenge to handle content-preserving operations—manipulations that do not manipulate the content. The well-known problem of "friendly attacks" occurs here as in any watermarking scheme: some signal manipulations must be allowed without breaking the watermark. In our case, every editing process that does not change the content itself is a friendly attack. Compression, dynamics, A/D-D/A-conversion, and many other operations that only change the signal but not the content described by the signal should not be detected. The idea is to use content information as an indicator for manipulations. The main challenge is to identify audio features appropriate to distinguish between content-preserving and content-changing manipulations.

Figure 3 shows the verification process of our content-fragile watermarking approach. We divide the audio file into frames of n samples. From these n samples, the feature checksums and the embedded watermark are retrieved and compared at the integrity check. As audio files are often cut, a resynchronization function is necessary to find the correct starting point of the watermark corresponding with the features. Our watermarking algorithm is robust against cropping attacks, but cutting out samples can lead to significant differences between extracted watermark and retrieved features. Therefore, a sync compare function tries to resynchronize both (features and watermark) if the integrity check is negative. Only if this is not successful, an integrity error is prompted.

2.2. *Digital audio features*

Extracted audio features are used to achieve robustness against allowed media manipulations while still being able to detect content manipulations. We want to ignore content-preserving operations which would lead to false alarms in cryptographic solutions and only identify real changes in the content. Additionally, we need to produce a binary representation of the audio content that is small enough to be embedded as a watermark and detailed enough to identify changes.

To produce a robust description of sound data, we have to examine which features of sound data can be extracted and described. Research has addressed this topic in psychoacoustics, for example, [20], and automated scene detection for videos, as in [20, 21]. We use the RMS, zero-crossing rate (ZCR), and the spectrum of the data as follows.

(i) RMS provides information about the energy of a number of samples of an audio file. It can be interpreted

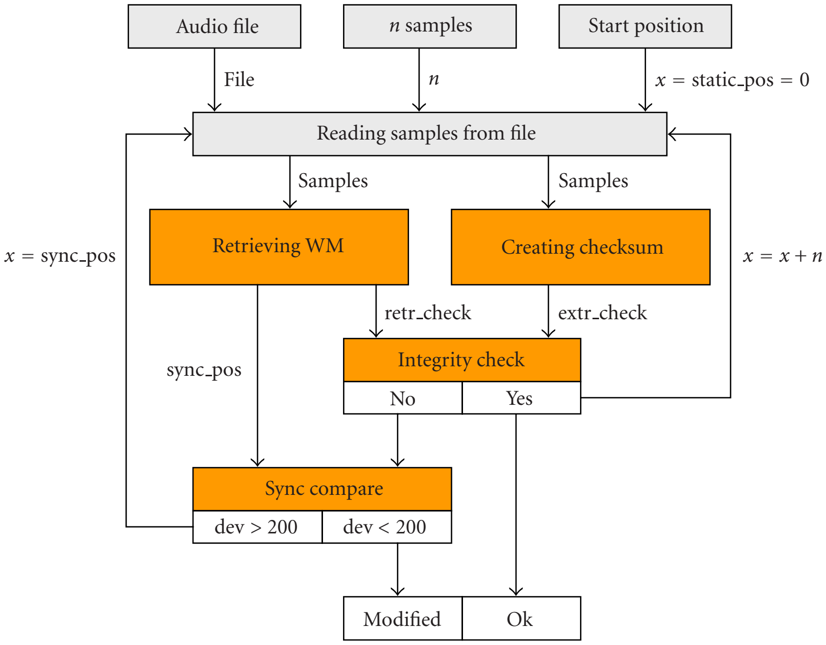

Figure 3: Content-fragile watermarking-based integrity decision.

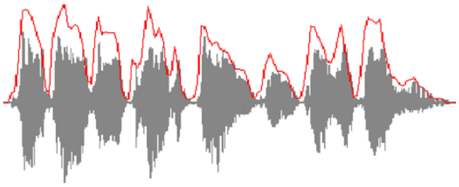

Figure 4: RMS curve of a speech sample.

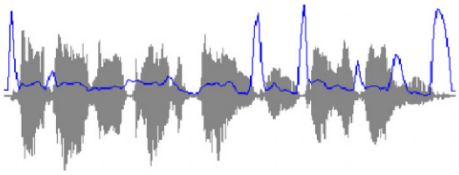

Figure 5: ZCR curve of a speech sample.

as loudness. If we can embed RMS information in a file and compare it after some attack, we can recognize muted parts or changes in the sequence (see Figure 4).

(ii) ZCR provides information about the amount of high frequencies in a window of sound data. It is calculated by counting the time the sign of the samples changes. The brightness of the sound data is described by it. Parts with small volume often have a high ZCR as they consist of noise or are similar to it (see Figure 5).

(iii) The transformation from time domain to frequency domain provides the spectrum information of audio data (see Figure 6). Pitch information can be retrieved from the spectrum. The amount of spectral information data is similar to the original sample data. Therefore, concepts for data reduction, like combining frequencies into subbands or quantization, are necessary.

To protect the semantic integrity of audio data, usually only a part of its full spectrum is required. For our approach, we choose a range similar to the frequency band transmitted with analogue telephones, from 500 Hz to 4000 Hz. Thereby, all information to detect changes in the content of spoken language is kept while other frequencies are ignored and the amount of data for the describing features is much less than the described audio. But even the amount of the thereby reduced data is too large for embedding. The maximum payload of today's watermarking algorithms is still too small. Therefore, to directly embed content descriptions, we have to use summaries of features or very global features—like the

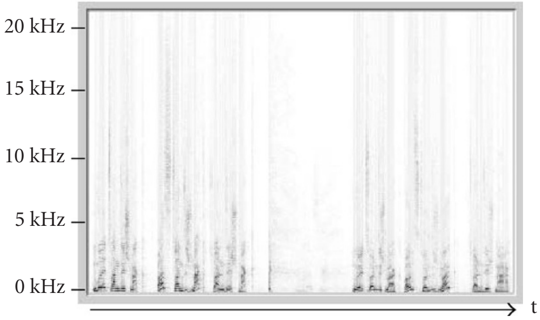

Figure 6: Spectrum of eight seconds of speech.

Table 1: Required bit rates for feature embedding.

FFT size	Features	Detail	Sync bits	Bit rate
Samples	#	Bit/feature	Bit	Bit/s
1.024	4	8	4	6,201.56
2.048	4	8	4	3,100.78
4.096	4	8	4	1,550.39
10.240	4	4	4	344.53
51.200	4	4	4	68.91
81.920	4	4	4	43.07

Table 2: Feature checksums based on different algorithms.

Window size		Key size	Sync bits	Type	Bit rate
Seconds	Samples	Bit	Bit		Bit/s
15.3252	675,840	160	4	SHA	10.7
12.3066	542,720	128	4	MD5/MAC	10.7
3.3669	148,480	32	4	CRC32	10.7
1.8576	81,920	16	4	CRC16	10.8
1.1146	49,152	8	4	XOR	10.8
0.7430	32,768	4	4	XOR	10.8

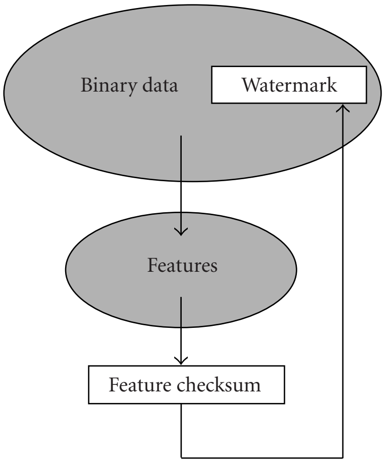

Figure 7: Feature checksums reduce the amount of embedded data.

RMS of one second of audio. This leads to security problems. As we only have information about a complete second, parts smaller than a second could be changed or removed without the possibility of localization. One cannot trust the complete second regardless the amount and position of change. It will also be a major challenge to disable possible specialized attacks trying to keep the overall feature the same while doing small but content-manipulating changes.

Table 1 shows a calculation of theoretically required watermarking algorithm bit rates. Here we extract four features (e.g., ZCR, RMS, and two frequency bands) and encode them with 8 or 4 bits. Quantization of the feature values is necessary to use a small number of bits. It also increases the feature robustness: less different values yield more robust ones against small changes. Quantization will set both original feature and modified feature to the same quantized value. We use quantization steps from 0.9 to 0.01. These are incremental values stepping from 0 to 1. If 0.9 is used, only one step is present, and basically no information regarding the feature is provided. With quantizer 0.01, 100 steps from 0 to 1 are made. The algorithm can differentiate between 100 values for feature representation.

Additionally, sync bits are required for resynchronization. This leads to very high bit rates at small FFT window sizes. Using big windows and low resolution reduces the required bit rates to about 43 bps. We could embed a content description about 5 times per second. But as 43 bps are still a rather high payload for current audio watermarking, robustness and transparency are not satisfactory. This leads to high error rates at retrieval and therefore to high false error rates. Our prototypic audio watermarking algorithm offers a bit rate of up to 30 bps if no strong attacks are to be expected, which would be the case in manipulation recognition scenarios. But with this average to high bit rate, compared to other algorithms available today, not only does robustness decrease but also error rates increase. Very robust watermarking algorithms today offer about 10 bits down to 1 bps.

2.3. *Feature checksums*

To circumvent the payload problem, we use feature checksums. We do not embed the robust features but only their checksum. Figure 7 illustrates this concept. The checksums can be compared to the actual media features checksums to detect content changes. An ideal feature is robust to all allowed changes—the checksum would be exactly the same after the manipulation. As we employ a sequence of features in every window, we need additional robustness: quantization reduces the required amount of bits and, at the same time, increases robustness as it maps similar values to the same quantized value. In Table 2, we list a number of checksums like hash functions (SHA, MD5), cyclic redundancy checks, or simple XOR functions. For hash functions, a certain amount of bits is required, therefore we can only work with big window sizes or a sequence of frames. XOR functions offer small window sizes. We can embed a feature checksum in less than a second with a bit rate of 10.8 bps into a single channel of CD quality PCM audio.

2.4. *Test results*

We use a prototypic implementation based on our own prototypic watermarking algorithm which uses spread spectrum and statistical techniques, different feature extractors, a feature comparison algorithm, and a feature checksum generator to evaluate our content-fragile watermarking concept. The basic idea of our tests can be described in the following steps.

(1) Select an audio file as a cover to be secured.
(2) Select one or more features describing the audio file.
(3) Retrieve the features for a given amount of time.
(4) Create a feature checksum.

Table 3: Embed/retrieve comparison for 4-bit RMS.

: Mode: Embed Bits per checksum: 4 bit Bit rate: 5.3833 bps Frames per checksum: 48 frames Included features: RMS in frequency domain 2000–6000 Hz			: Mode: Retrieve Bits per checksum: 4 bit Bit rate: 5.3833 bps Frames per checksum: 48 frames Included features: RMS in frequency domain 2000–6000 Hz				
					Checksum		
No	Time (min:s)	Checksum	No	Time (min:s)	Extr	Retr	Integrity
0	0:0	11	0	0:3.72719	11	12	modified
1	0:2.22912	9	1	0:3.71299	9	4	modified
2	0:4.45823	7	2	0:6.68261	7	12	modified
3	0:6.68735	13	3	0:12.6269	13	12	modified
4	0:8.91646	9	4	0:11.1427	9	0	modified
5	0:11.1456	3	5	0:12.6313	3	4	modified
6	0:13.3747	7	6	0:13.3739	7	7	ok
7	0:15.6038	12	7	0:15.6022	12	12	ok
8	0:17.8329	9	8	0:17.8322	9	9	ok
9	0:20.062	3	9	0:20.0586	3	3	ok
10	0:22.2912	8	10	0:23.775	8	4	modified
11	0:24.5203	4	11	0:24.5186	4	4	ok
12	0:26.7494	1	12	0:26.7523	1	1	ok
13	0:28.9785	12	13	0:28.9769	12	12	ok
14	0:31.2076	15	14	0:32.6924	15	8	modified
15	0:33.4367	11	15	0:37.8899	11	12	modified
16	0:35.6659	14	16	0:37.1661	14	12	modified
17	0:37.895	14	17	0:37.9079	14	14	ok
18	0:40.1241	0	18	0:40.1361	0	0	ok
19	0:42.3532	7	19	0:42.3515	7	7	ok
20	0:44.5823	4	20	0:44.5798	4	4	ok
Embed mode: Checksums are generated and embedded as a watermark			Retrieve mode: Checksums are generated and compared to those retrieved as a watermark				

(5) Embed the feature checksum as a watermark.
(6) Attack the cover.
(7) Retrieve the watermark from the attacked cover.
(8) Retrieve the features from the attacked cover and generate the checksums.
(9) Compare both to decide if a content-change has occurred.

Table 3 shows an example where a 4-bit checksum and two sync bits are embedded every 48 frames. In the left row, the embedded feature checksums are presented and in the right row, the results of a retrieve process. The comparison includes actual extracted features, retrieved features, and a decision if integrity has been corrupted. In this example, we see that extracted feature checksums after embedding and retrieval are matching, while the extracted watermark shows other features. This may seem confusing at first sight as one would assume the embedded information and the extracted features in embed mode to be similar. In this example, the chosen watermarking parameters are too weak and produce bit errors at retrieval but at the same time do not influence the robust features. It is clear that an optimal trade-off between the robustness and transparency of the watermark will provide the best results.

Audio watermarking algorithms are usually not completely reliable regarding the retrieval of single embedded bits. Certain number of errors in the detected watermarks can be expected and compensated by error-correction codes and redundancy. But as the data rate of the watermarking algorithms is already low without these additional mechanisms, content-fragile watermarks cannot rely on error compensation. Therefore, to achieve good test results, watermarking and feature parameters have to be chosen carefully to prevent a high error rate. In Figure 8, a set of optimized

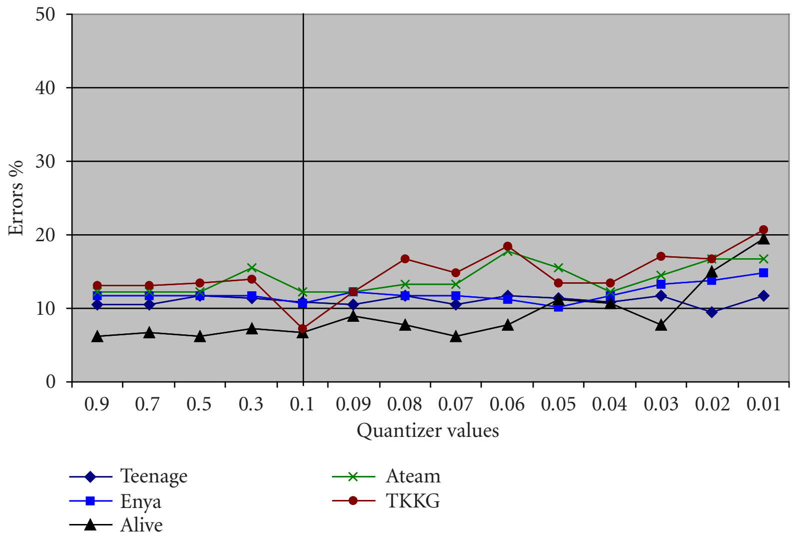

Figure 8: Optimized parameters lead to error rates below 20% (RMS, checksum 4 bit).

parameters has been identified and tested with five audio files ranging from rock music to radio drama. RMS is chosen as the extracted feature. To receive optimal results, we keep a certain distance between the frequency band the watermark is embedded in and the feature it is extracted from. In this example, the feature band is 2 kHz to 6 kHz. The watermark is embedded in the band from 10 kHz to 14 kHz.

Even with these optimized parameters, for the retrieval of feature checksums, a false error rate between 5% and 20% is usual. Today's audio watermarking algorithms offer error rates of 1% or less per embedded bit. This adds up to a bigger error rate in our application as one wrong bit in the multibit checksum results in an error. For common audio watermarking applications, a 5% error rate for embedded watermarks is acceptable. Both, the error rate of the watermarking algorithm and the possibility of changing the monitored feature by embedding the watermark, sum up to a basic error rate, which is detected even if no attacks have occurred. This basic error rate has to be taken into account when a decision regarding the integrity of the audio material is made.

As already stated in Section 2.2, quantizer sizes influence robustness. For the results in Figure 8, a quantizer value of 0.9 basically means that all features are identified by the same value, while 0.01 provides a detailed representation. Error rates increase with the level of detail.

In Figure 9, we show test results after performing a stirmark benchmark audio attack [22] for the parameter RMS. We embed a feature vector with the parameters of Figure 8 and run a number of audio manipulations of different strength on the marked file. Then the watermark is retrieved and both the retrieved and the recalculated feature vectors are compared.

The content-preserving attacks “normalize,” “invert,” and “amplify” result in equal error rates as in the no-operation attack “nothing” or after only embedding the watermark. An error rate below 20% can be seen as a threshold for content-preserving operations. Content manipulations like filters (lowpass, highpass), the addition of noise (addnoise) or humming (addbrumm), and removal of samples result in higher error rates up to almost 100%. The different quantization values have a significant influence on the error rate again, but the behavior is the same for all attack types: a lower resolution results in lower error rates.

While these attacks may be assumed to be content preserving in some cases, for example, lowpass filtering common in audio transmission, the results show that a certain discrimination between attacks is possible. The results also correspond to the attack strength. Lower noise values lead to lower error rates.

The test results are encouraging. A threshold may be necessary to filter an unavoidable error level of about 20%, but attacks can be identified. Quantization values can be used as a fragility parameter. A similar behavior is observed in different audio files including speech, environmental recordings, and music, making this approach useful for various applications.

3. INVERTIBLE AUDIO WATERMARKING

Based on the general idea of invertible watermarking, an invertible scheme for audio has to combine a lossless compression with different cryptographic functions, see Figure 10. An audio stream consists of samples with variable numbers of bits describing one sample value. We take a number of consecutive samples and call them a frame. Now one bit layer of this frame is selected and compressed by a lossless compression algorithm. For example, we would build a frame of 10 000 16-bit samples and take bit #5 from each sample. The difference between memory requirements of the original and the compressed bit layer can now be used to carry additional security information. In our example, the compressed 10 000 bits of layer #5 could require only 9 000 bits to represent. The

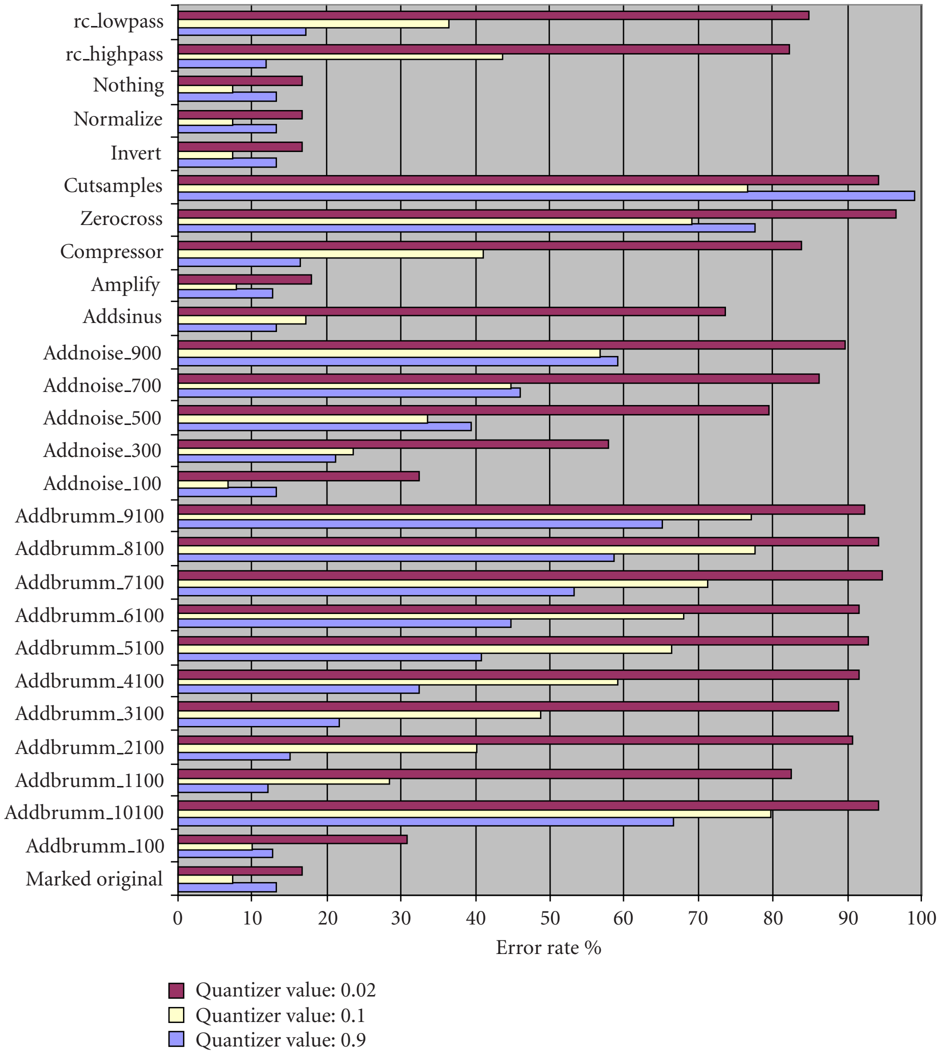

Figure 9: Stirmark audio test results. Stronger attacks lead to higher error rates (RMS, checksum 4 bit).

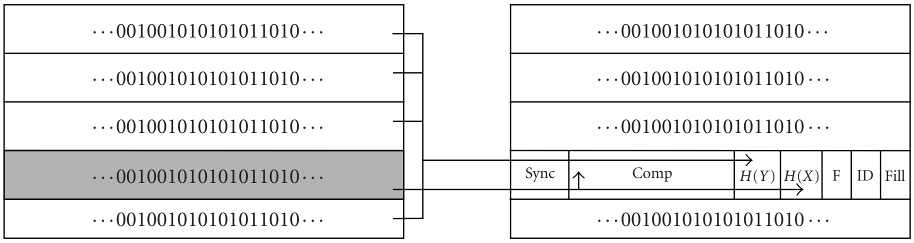

Figure 10: Invertible audio watermarking. The bits of one bit layer are compressed and the resulting free space is used to embed additional security information.

resulting 1 000 bits can be used as security information like, for example, a hash of the other 15 bit layers. The original bit vector is replaced by the compressed bit vector and the security information. As the complete information about the original bit layer is still available in compressed form, it can be decompressed at any time, and by overwriting the new information with the original bits, we get the original frame back.

3.1. Invertible authentication for audio streams

As discussed, today's invertible watermarking solutions are only available for image data. Here, only one complete image

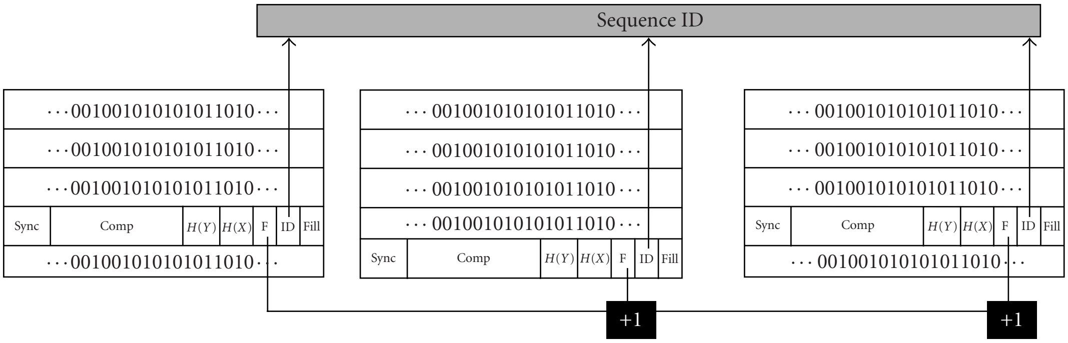

Figure 11: Audio watermarking requires stream synchronization to allow cutting the material. Therefore, an incremental frame ID is included in the watermark.

will be protected. If the same concept is transferred to audio data, certain problems arise: the amount of data for a long recording can easily become more than one GB. If the invertible watermark is to be embedded in a special device, very large memory reserves would be necessary. Besides this technical problem, integrity may not be lost even if the original data is not completely present. A recording of an interview may be edited later, removing an introduction of the reporter or some regards at the end. The message of the interview will not be corrupted if this information is removed.

Therefore, we suggest a frame-based solution like that introduced in Section 2. A number of consecutive samples are handled as an audio frame, for example, 44100 samples for one second of CD-quality mono data. This frame F_i is now protected like, for example, a single image. It includes all necessary information to prove its integrity. But additional security information and a synchronization header are necessary as shown in Figure 11.

(i) A sequence ID_S is embedded in every frame F_i. It verifies that the frame belongs to a certain recording. This provides security versus exchanges from other recordings. If ID_S is not included, an attacker could overwrite a part of the protected audio data with a different but also protected stream from another recording but from the same position without being detected.

(ii) An incremental frame ID_T is also embedded. This provides security against exchanges in the sequence of frames F_i of the protected sequence. Swapping a number of frames would not be detectable without this ID and would lead to manipulation possibilities.

(iii) A synchronization header ($\text{Data}_{\text{sync}}$) is also necessary. Otherwise, cutting the audio data would usually lead to a complete loss of the security information, as the correct start position of a new frame would be undetectable. With the help of the synchronization header, the algorithm can scan for a new starting point if it detects a corrupt frame.

With these three additional mechanisms, invertible audio watermarking becomes more usable for many applications as a number of attacks are disabled and a certain amount of editing is allowed. A tool for integrity verification could identify gaps in the sequence and inform the user about it. This user then can decide whether to trust the information close to the gaps or not as, for example, a third party could have removed words from a speech. From our discussion in Section 1.3, the protocol for audio data can be written for each audio frame F_i ($i = 1\ldots$ number of audio frames) as follows:

$$\begin{aligned}F_iW &= F_{i\,\text{remaining}}//\text{Data}_{\text{sync}}//\text{Data}_{\text{info}}//W_i//\text{Data}_{\text{fill}},\\ W &= E_{\text{AES}}\left(E_{\text{AES}}\left(\text{C}_{\text{AudioLayerBits}}, k_H\left(F_{i\,\text{remaining}}\right)\right), K_{\text{secret}}\right)\\ &\quad //\text{HMAC}\left(\text{AudioLayerBits}, K_{\text{secret}}\right)\\ &\quad //\text{RSA}_{\text{signature}}\left(H\left(F_{i\text{-remaining}}\right.\right.\\ &\quad //E_{\text{AES}}\left(E_{\text{AES}}\left(\text{C}_{\text{AudioLayerBits}}, k_H\left(F_{i\,\text{remaining}}\right)\right), K_{\text{secret}}\right)\\ &\quad \left.\left.//\text{HMAC}\left(\text{AudioLayerBits}, K_{\text{secret}}\right)//\text{Data}_{\text{Sync}}\right)\right),\end{aligned}$$

where AudioLayerBits is the bit vector to be replaced in F_i, and $F_{i\,\text{remaining}}$ is the set of the remaining bit vectors in F_i.

3.2. Compression techniques and capacity evaluation

Based on the general invertible concepts, the next major question is how to perform a lossless audio compression C to achieve invertibility: to get back the exact original of the audio representation, common audio compressions like mp3 are not acceptable due to their lossy characteristics. Compression schemes also applied to text or software, like the common zip compression C, satisfy this requirement but are far less efficient than lossy audio compression.

Therefore, we design the following compression algorithm.

(1) The required number of bits r and the number of samples n to be used as one frame are provided as parameters.

(2) From the n samples, each lowest bit is added to a bit vector B of the length n.

(3) B is compressed by a lossless algorithm, producing a compressed bit vector B′ of length n'.

(4) If $n - n' < r$, the compression of the bit layer is not sufficiently efficient and the next higher bit layer is compressed.

(5) If we win r bits from the compression, the current bit layer becomes the one we embed the information into for this frame.

(6) If even at bit layer #15 the compression is not sufficiently efficient, embedding is not possible in this frame.

Table 4 shows an example of this process. The parameters are 44100 samples for one frame and a requirement of 2000 bits. In the first frame at bit #0, we already receive good compression results. The difference $n - n'$ is 3412, more than required. In the second frame, bits #0 to #7 do not provide a positive compression ratio, so bit #8 is selected as the compressed layer. To identify the chosen bit layer, a synchronization sequence embedded into the $\text{Data}_{\text{info}}$ is necessary, identifying the compressed layer for every frame.

In Table 5, we provide a comparison of capacities of four example files A to D for two frames of 44100 samples. A first assumption about bit requirements can be made based on the knowledge about required components. As multiple hash functions are available and the length of the RSA signature is key dependent, the capacity requirements are calculated as follows:

(i) sync info, for example, 64 bit;

(ii) two hash values:
 (a) remaining audio information, for example, 256 bit,
 (b) selected bit layer, for example, 256 bit;

(iii) RSA digital signature, for example, 512 bit;

(iv) compressed bits are encrypted by a symmetric key scheme (AES), that is, adding max. 63 bits.

This sums up to about 1100 bits. Therefore, any compression result providing 1100 bits of gain would be suitable for embedding invertible security information. In the example of Table 5, in frame 1, the information will be embedded in bit layer #8 of file A and in layer #0 of B, C, and D. In frame 2, A and C require bit layer #8, while B and D can use bit layer #0. An important observation is the fact that the capacity of compression results is not always increasing as one would assume when looking at the examples for still images of [9]. In frame 2 of Table 5, column D, the amount of received bits decreases from bit layer #0 to #7 and then becomes a constant value for bit layers #8 to #15. Quantization, changes of bit representation, and addition of noise are possible reasons for this effect.

4. APPLICATIONS

Content security for digital audio is not discussed today as much as for images or video data. In this section, we discuss a selection of possible scenarios where either content-fragile or invertible watermarking schemes like the ones we described in Sections 2 and 3 will become necessary.

4.1. News data authentication

Digital audio downloads on the Internet can replace radio news. Interviews and reports will be recorded, digitized, and uploaded to news servers. With content-fragile watermarking, the trust in the information can be increased. The source of the news, for example, a reporter or even the recording device, embeds a content-fragile watermark into the audio data and encrypts the content information with a private key. Now everybody who uses a corresponding detection algorithm would be able to verify the content. If the watermarking keys were distributed freely, only the public key of the embedding party would be required for verification.

The robustness of the algorithm to content-preserving operations, for example, format changes or volume changes, allows the news distributor to adjust the data to his common format without the need of a new verification process. Only the source of the data has to be trusted; all changes in the distribution chain will be detected. A person receiving the news

Table 4: Compression efficiency changes from frame to frame.

Bit#	Bits/Orig.	Bits/Comp.	Difference	Comp. Factor
		Frame 1		
0	5513	2101	**3412**	0.381
1	5513	2307	3206	0.418
2	5513	2517	2996	0.457
3	5513	3424	2089	0.621
4	5513	4415	1098	0.801
5	5513	5225	288	0.948
6	5513	5574	−61	1.011
7	5513	5603	−90	1.016
8	5513	1299	4214	0.236
9	5513	1298	4215	0.235
10	5513	1298	4215	0.235
11	5513	1298	4215	0.235
12	5513	1301	4212	0.236
13	5513	1386	4127	0.251
14	5513	1605	3908	0.291
15	5513	1859	3654	0.337
		Frame 2		
0	5513	5345	168	0.970
1	5513	5566	−53	1.010
2	5513	5599	−86	1.016
3	5513	5603	−90	1.016
4	5513	5603	−90	1.016
5	5513	5606	−93	1.017
6	5513	5602	−89	1.016
7	5513	5600	−87	1.016
8	5513	1725	**3788**	0.313
9	5513	1726	3787	0.313
10	5513	1726	3787	0.313
11	5513	1726	3787	0.313
12	5513	1742	3771	0.316
13	5513	2626	2887	0.476
14	5513	3611	1902	0.655
15	5513	4647	866	0.843

Table 5: Capacity comparison of four example files and two frames.

Frame	Bit#	A	B	C	D	Frame	Bit#	A	B	C	D
1	0	0	5362	1702	3793	2	0	0	5097	0	2708
1	1	0	5362	1690	3792	2	1	0	5098	0	2359
1	2	0	5362	1670	3793	2	2	0	5047	0	1666
1	3	0	5362	1648	3792	2	3	0	4946	0	772
1	4	0	5361	1630	3779	2	4	0	4861	0	144
1	5	0	5362	1631	3708	2	5	0	4801	0	0
1	6	0	5347	1625	3584	2	6	0	4715	0	0
1	7	0	5347	1630	2395	2	7	0	4701	0	0
1	8	3113	5362	4195	3792	2	8	3151	5097	4051	2746
1	9	3114	5361	4195	3793	2	9	3151	5097	4051	2746
1	10	3113	5362	4128	3792	2	10	3151	5097	3987	2746
1	11	3113	5362	3567	3792	2	11	3151	5098	3114	2746
1	12	2998	5362	2941	3793	2	12	2982	5097	2118	2746
1	13	2081	5361	2328	3792	2	13	1984	5097	1116	2746
1	14	1115	5362	1844	3793	2	14	1027	5097	276	2746
1	15	128	5362	1741	3792	2	15	84	5097	0	2746

therefore can be sure the content is not censored or manipulated by third parties.

4.2. *Surveillance recordings*

Surveillance recordings are most often made by cameras. Recently, requirements regarding the trustworthiness of digital versions of such cameras became an important issue. If the recorded content is easily manipulated, the concept of surveillance and its weight at court is flawed. With digital audio content authentication, the audio channels of surveillance recordings can be protected.

Invertible methods would be applied in scenarios where a high security is required and the audio data is not compressed but stored directly onto high-capacity mediums. The watermark would act like a seal that will be broken if manipulations take place. At the same time, the inversion option enables a selected group of persons to work with original quality if necessary.

Content-fragile watermarking will be applied if operations like compression are forecasted and accepted after embedding. The robust watermark applied in this approach will survive the compression algorithm and provide the content information at a later time to verify the integrity of the recording.

4.3. *Forensic recordings*

The assumption that an audio file has to be highly secured can be made with forensic recordings. When such a recording is performed, for example, of an interview with witness, protection of the content is very important and our invertible approach can be used to ensure data authentication. Usually the invertible aspect will not be required as the spoken language is not very fragile against bit changes in the lower layers. Invertibility is important when very small changes in the audio can have some effect. This may be the case when a digital copy of an analogue recording is made, and later based on the digital copy, assumptions about cuts in the analogue media shall be made. Now the addition of noise from the watermark may mask the slight changes one can perceive at the cutting points. The possibility of setting back the recording to its original state is an important increase of usability here. A similar case is the detection of environmental noises in telephone calls, for example. A marked recording with added noise will also make it hard to estimate the nature of the background noise as both types of noise will mix.

4.4. *CD-master protection*

In the examples given above, only speech or environment information is protected. But music can also be the subject of our protection schemes; CD masters are valuable pieces of audio data, which also require an exact reproduction to ensure copies of high quality. Our invertible audio watermarking scheme offers two valuable mechanisms for this scenario. When the CD master is protected by an invertible watermark, it can be sent without any additional security requirements via mail or internet. Any third party capturing the copy and not possessing the secret key can get an idea how the CD will sound like, but audio quality is too low for illegal reproduction. After the CD arrives at its destination, the CD copy plant will use the sender's public key to verify the integrity of the audio tracks and the previously exchanged secret key to remove the watermark. Thereby an error-proof and secure copy has been transmitted via an insecure environment.

5. SUMMARY AND CONCLUSION

In this paper, we introduce two concepts for digital audio content authentication. Content-fragile watermarking is based on combining robust watermarking and fragile content features. It is suitable for applications where certain operations changing the binary representation of the content are acceptable. The robust nature of the watermark and the right choice of content features and their quantization provide tolerance to such operations while they still enable us to identify content changes. The invertible watermarking approach is suited for high-security scenarios. It offers no robustness to any kind of operation but cutting. Our frame-based approach allows the detection of cuts and the resynchronization afterwards. The verification of integrity is much more exact than in the content-fragile approach since cryptographic hash functions are applied. An important additional feature is the invertibility allowing recreating the original state of the data if the corresponding secret key is present.

We provide test results for both authentication schemes. Content-fragile watermarking error rates increase with the strength of attacks. Therefore, a threshold-based identification of content changes depending on the application is possible. One source of false alarms in this approach is errors in the retrieved watermark. Improving the watermarking algorithm will decrease false alarms. Both, a better transparency to reduce the effects of the embedded watermark on the retrieved features and a more reliable watermarking detection would decrease the basic error rate.

The test results of the invertible approach address mainly compression results. We show that a flexible detection of suitable bit vectors from frame to frame is necessary to achieve an optimal trade-off between quality and compression. We also prove the general possibility to embed a suitable amount of security data. The compression rates of the audio bits are suitable to carry all required information in all examples.

The security of both introduced approaches depends on keys. A secure watermarking algorithm like the one applied to embed the content-fragile information in Section 2 will always be based on a secret user key. One can assume that the basic embedding function will become known to the public sooner or later, so security based on secret algorithms will lead to serious security risks. The invertible approach in Section 3 includes two key management methods. The compressed data is encrypted with a secret key scheme while the verification of integrity is based on a public key scheme. Therefore, key management is an important issue for both approaches.

Content-fragile watermarking requires at least a key distribution concept for the watermark key that will be application dependent. The key could be distributed to every interested party so everyone can verify the integrity of the content, or the key is present at a trusted third party where the marked material can be uploaded for verification. If our suggestion to use asymmetric encryption to add authentication of the embedding party of the content-fragile data is applied, a PKI is necessary.

Invertible watermarking also requires a PKI for integrity verification as the hash values of the original data are encrypted with an asymmetric scheme. Secure key exchange between those parties that should be able to decrypt the compressed data and set back the audio data to its original state is also necessary.

To conclude, we see that audio content security is an important new domain in audio processing and watermarking as well as in general audio research. Our paper shows up different promising directions. Future work in content-fragile audio watermarking concentrates on further feature extraction and development of a demonstrator software to finally achieve a complete framework. Current test results show the general correctness of our ideas but also identify the necessity of further research.

ACKNOWLEDGMENT

We would like to thank Patrick Horster from University Klagenfurt (Austria) for his input regarding formalizations of the invertible watermarking scheme.

REFERENCES

[1] J. Dittmann, P. Wohlmacher, and K. Nahrstedt, "Using cryptographic and watermarking algorithms," *IEEE Multimedia*, vol. 8, no. 4, pp. 54–65, 2001.

[2] J. Dittmann, M. Steinebach, T. Kunkelmann, and L. Stoffels, "H2O4M - watermarking for media: Classification, quality evaluation, design improvements," in *Proc. 8th ACM International Multimedia Conference (ACM Multimedia '00)*, pp. 107–110, Los Angeles, Calif, USA, November 2000.

[3] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on copyright marking systems," in *Proc. 2nd International Workshop on Information Hiding (IHW '98)*, vol. 1525 of *Lecture Notes in Computer Science*, pp. 219–239, Springer-Verlag, Portland, Ore, USA, April 1998.

[4] E. Lin and E. J. Delp, "A review of fragile image watermarks," in *Proc. ACM International Multimedia Conference (ACM Multimedia '99)*, J. Dittmann, K. Nahrstedt, and P. Wohlmacher, Eds., pp. 25–29, Orlando, Fla, USA, October–November 1999.

[5] J. Fridrich, "Methods for tamper detection of digital images," in *Proc. ACM International Multimedia Conference (ACM Multimedia '99)*, J. Dittmann, K. Nahrstedt, and P. Wohlmacher, Eds., pp. 29–34, Orlando, Fla, USA, October–November 1999.

[6] J. Dittmann, M. Steinebach, I. Rimac, S. Fischer, and R. Steinmetz, "Combined video and audio watermarking: embedding content information in multimedia data," in *Security and Watermarking of Multimedia Contents II*, vol. 3971 of *SPIE Proceedings*, pp. 455–464, San Jose, Calif, USA, January 2000.

[7] J. Dittmann, "Content-fragile watermarking for image authentication," in *Security and Watermarking of Multimedia Contents III*, vol. 4314 of *SPIE Proceedings*, pp. 175–184, San Jose, Calif, USA, January 2001.

[8] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding—new paradigm in digital watermarking," *Journal on Applied Signal Processing*, vol. 2002, no. 2, pp. 185–196, 2002.

[9] J. Dittmann, M. Steinebach, and L. Croce Ferri, "Watermarking protocols for authentication and ownership protection based on timestamps and holograms," in *Photonics West 2002, Electronic Imaging: Science and Technology; Multimedia Processing and Applications, Security and Watermarking of Multimedia Contents IV*, E. J. Delp and P. W. Wong, Eds., vol. 4675 of *SPIE Proceedings*, pp. 240–251, San Jose, Calif, USA, January 2002.

[10] M. Arnold, "Audio watermarking: Features, applications and algorithms," in *Proc. IEEE International Conference on Multimedia and Expo (ICME '00)*, pp. 1013–1016, New York, NY, USA, July–August 2000.

[11] J. D. Gordy and L. T. Bruton, "Performance evaluation of digital audio watermarking algorithms," in *Proc. 43rd IEEE Midwest Symposium on Circuits and Systems (MWSCAS '00)*, pp. 456–459, Lansing, Mich, USA, August 2000.

[12] D. Kirovski and H. Malvar, "Spread-spectrum audio watermarking: Requirements, applications, and limitations," in *IEEE 4th Workshop on Multimedia Signal Processing*, Cannes, France, October 2001.

[13] P. Nintanavongsa and T. Amornkraksa, "Using raw speech as a watermark, does it work?," in *Proc. International Federation for Information Processing Communications and Multimedia Security (CMS), Joint Working Conference IFIP TC6 and TC11*, R. Steinmetz, J. Dittmann, and M. Steinebach, Eds., Kluwer Academic, Darmstadt, Germany, May 2001.

[14] C. Neubauer and J. Herre, "Audio watermarking of MPEG-2 AAC bit streams," in *AES 108th Convention*, Porte Maillot, Paris, France, February 2000.

[15] L. Boney, A. H. Tewfik, and K. N. Hamdy, "Digital watermarks for audio signals," in *VIII European Signal Proc. Conf. (EUSIPCO '96)*, Trieste, Italy, September 1996.

[16] J. Dittmann, *Digitale Wasserzeichen*, Springer-Verlag, Berlin, Heidelberg, 2000.

[17] C.-P. Wu and C.-C. J. Kuo, "Comparison of two speech content authentication approaches," in *Photonics West 2002: Electronic Imaging, Security and Watermarking of Multimedia Contents IV*, vol. 4675 of *SPIE Proceedings*, pp. 158–169, San Jose, Calif, USA, January 2002.

[18] J. Dittmann, M. Steinebach, L. Croce Ferri, A. Mayerhöfer, and C. Vielhauer, "Advanced multimedia security solutions for data and owner authentication," in *Applications of Digital Image Processing XXIV*, vol. 4472 of *SPIE Proceedings*, pp. 132–143, San Diego, Calif, USA, July–August 2001.

[19] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Photonics West 2001: Electronic Imaging, Security and Watermarking of Multimedia Contents III*, vol. 4314 of *SPIE Proceedings*, pp. 197–208, San Jose, Calif, USA, January 2001.

[20] Z. Liu, J. Huang, Y. Wang, and T. Chen, "Audio feature extraction & analysis for scene classification," in *IEEE 1st Workshop on Multimedia Signal Processing*, pp. 343–348, Princeton, NJ, USA, June 1997.

[21] S. Pfeiffer, *Information Retrieval aus digitalisierten Audiospuren von Filmen*, Shaker Verlag, Aachen, Germany, 1999.

[22] M. Steinebach, F. Petitcolas, F. Raynal, et al., "StirMark benchmark: Audio watermarking attacks," in *Proc. International Conference on Information Technology: Coding and Computing (ITCC '01)*, pp. 49–54, Las Vegas, Nev, USA, April 2001.

Martin Steinebach is a Research Assistant at Fraunhofer IPSI (Integrated Publication and Information Systems Institute). His main research topic is digital audio watermarking. Current activities are watermarking algorithms for mp2, MIDI and PCM data, feature extraction for content-fragile watermarking, attacks on audio watermarks, and concepts for applying audio watermarks in eCommerce environments. He studied computer science at the Technical University of Darmstadt and finished his Diploma thesis on copyright protection for digital audio in 1999. Martin Steinebach was the Organizing Committee Chair of CMS 2001 and coorganized the Watermarking Quality Evaluation Special Session at ITCC International Conference on Information Technology: Coding and Computing 2002. Since 2002 he is the Head of the Department MERIT (Media Security in IT) and of the C4M Competence Center for Media Security.

Jana Dittmann has been a Full Professor in the field of multimedia and media security at the Otto-von-Guericke University Magdeburg since September 2002. She studied computer science and economy at the Technical University in Darmstadt and worked as a Research Assistant at the GMD-IPSI (later Fraunhofer IPSI) from 1996 to 2002. In 1999, she received her Ph.D. from the Technical University of Darmstadt. At IPSI, she was one of the founders and the leader of the C4M Competence Center for Media Security. Jana Dittmann specializes in the field of Multimedia Security. Her research has mainly focused on digital watermarking and content-based digital signatures for data authentication and for copyright protection. She has many national and international publications, is a member of several conference PCs, and organizes workshops and conferences in the field of multimedia and security issues. Since June 2002, she is Editor of the Editorial Board of ACM Multimedia Systems Journal. She was involved in the organization of all the last five Multimedia and Security Workshops at ACM Multimedia. In 2001, she was a Cochair of the CMS 2001 conference that took place in May 2002 in Darmstadt, Germany. Furthermore, she organized several special sessions, for example, on watermarking quality evaluation and on biometrics.