Journal of Information Hiding and Multimedia Signal Processing ©2012 ISSN 2073-4212
Ubiquitous International Volume **3**, Number **1**, January **2012**

Real-time Attacks on Audio Steganography

M. Nutzinger

Theobroma Systems Design und Consulting GmbH
1230 Vienna, Austria
marcus.nutzinger@gmail.com

Received May 2011; revised November 2011

Abstract. *Steganographic algorithms work in different ways to embed secret data into cover media. In this paper, techniques are presented which aim at the prevention of steganography usage in digital audio data, independent of the underlying algorithm. Differing from other approaches which try to detect embedded data or perform time-consuming computations as part of their process, we focus on a comprehensive steganography prevention which works in real-time on any audio data. In addition our techniques aim at minimizing the audio quality degradation, making them usable for different fields of application. Tests of our system show the effectiveness of the proposed techniques: The steganographic receiver is disturbed, resulting in increased bit error rates and rendering the steganographic channel unusable or at least limiting its usability. We also evaluate the resulting audio quality and compare our implementation to the StirMark for Audio benchmark. The results show that our apprach is different from this attack suite as we propose novel methods not available within StirMark for Audio and our emphasis is the preservation of the audio quality. The promising quality results as well as the prevention effectiveness justify the existance of our techniques as an enhancement to the StirMark for Audio benchmark.*
Keywords: Steganography, Attacks, Real-time Processing, Audio Quality Preservation.

1. **Introduction.** Steganography is the science which deals with hidden information exchange. While cryptography protects the content of secret data, the goal of steganography is to conceal the mere existence of secret information [23]. In technical steganography, digital data like images, audio and video files are used as cover media [36]. In our research, the focus has been set to audio steganography, where we mostly used Voice-over-IP (VoIP) communications as cover media. Because of the characteristics of the human auditory system (HAS), the application of steganography in this area is a challenging task. This is due to the fact that the HAS notices slight changes in the audio data more quickly than e.g. the human visual system (HVS) reacts on small distortions in pictures [4, 5]. Due to the sensitivity of the HAS, there are three important requirements for steganography when applied to audio data: Undetectability and inaudibility, robustness and capacity [2, 41].

A clear motivation for the aim to prevent steganographic communications is the fact that this technology can well be exploited to support organized crime [38]. Further, looking at methods for steganography prevention also leads to new approaches for the design of robust steganographic algorithms, which therefore justifies the research on prevention techniques. Also only little research has been done on this field in recent works (see Sect. 2).

We define several requirements for the prevention of steganography, where the important requirements for audio steganography have to be kept in mind when designing attack techniques. Related to the first requirement for audio steganography, all defense techniques shall be undetectable in a way that the audio quality is not degraded by the specific attack technique. Further, we dissociate this work from steganalysis. Steganalysis aims at the detection of embedded data present in a given cover medium. This process can be done through statistical analysis as well as more sophisticated techniques, dealing with specific parameters of the audio signal [11, 12]. In contrast, the presented attack techniques shall be applicable on all kinds of audio material, having the goal of interfering the steganographic receiver if there exists a steganographic communication. On the other hand, if no steganographic communication is present, the techniques can be applied as well. Due to the fact that no noticeable disturbances are introduced to the audio signal, this approach does not harm the underlying communication, e.g. a VoIP call.

The major contribution of this paper is a novel combination of various basic signal processing operations for the generic prevention of steganographic communications in audio cover media. While many of the existing approaches, dealing with steganographic prevention, perform some kind of steganalysis (see Sect. 2), our approach is different. As mentioned above, no evidence for the existence of embedded data is searched. This way the processing time is negligible, making it possible for our techniques to work in real-time environments, e.g. on a router between two VoIP clients. Further, the parameters of the attack techniques are chosen in such a way that the quality of the audio data is not negatively influenced. Therefore our method is usable on all kinds of audio streams, disturbing a possible steganographic receiver while keeping a reasonable audio quality.

This paper is organized as follows: In Sect. 2 related approaches, dealing with the prevention of steganographic communications, are highlighted and the distinction of our work is pointed out. Section 3 details our novel contributions. Each involved signal processing operation as well as its implementation and purpose in the overall system is outlined. Results from tests with our working prototype are presented in Sect. 4. This section shows the outcome of evaluating the quality of audio signals before and after application of the described techniques. In addition the efficiency of our techniques is tested against three steganographic algorithms. All results are further compared to the attack types from StirMark for Audio [8]. Ideas for improvements and future research conclude this paper.

2. **Related Work.**

Research related to this paper can be divided into three areas. On the one hand techniques have been proposed which aim at the attack of digital watermarking or steganography systems. In this area different attacks and attack suits have been evolved both for digital images and audio data as cover media. The second part are works which deal with the detection of embedded data. This area has been investigated in recent research and we will give a short overview on such approaches and their difference to our proposed techniques. Finally this section will also show methods to recover embedded data from media which has been damaged by some kind of attack. While such research is contrary to our goal it is still important to regard the approaches in this area, at least to improve the proposed methods.

2.1. **Attacks on Digital Watermarking and Steganography Systems.**

While [42] mentioned attacks on steganographic systems several years ago, they focused on digital images as cover media. Further their attacks were directed to certain tools performing steganography. As a consequence, no comprehensive prevention techniques were

described. In 1998 the first version of StirMark was published, a benchmark for watermarking systems which applies various signal processing modifications and attacks on the cover media in order to render the embedded data unusable [34, 35]. The first version of the StirMark benchmark [33] was built for image watermarking and steganography algorithms, hence [35] mainly focuses on digital images. However some notes on the introduction of jitter into audio data are given, a method which we took up and implemented it as "variable time delay" (see Sect. 3). After StirMark was released the initial researchers as well as other scientists improved the first version and in 2001 a StirMark benchmark for audio cover media was described in [40]. This work has been enhanced by the addition of lossy compression techniques like MP3 and Ogg [7]. In [8] an updated view on StirMark for Audio benchmarking is given and all implemented attacks are listed and described briefly. We will base our evaluation on this most recent paper and the software available at [39] (see Sect. 4).

[10] looks at the prevention of network steganography by an active warden. The goal of this research has been to stop all steganographic communications at a network firewall. While unstructured carriers like images and audio files are also mentioned, their focus lies on structured carriers like a TCP/IP communication where secret data is embedded in unused protocol header fields. For the actual prevention techniques [10] introduced the Minimal Requisite Fidelity (MRF). MRF denotes the level of destruction of the carrier so that it is still accepted by users while the embedded data is destroyed. Our work enhances the work from [10] by focussing on audio data, an unstructured carrier, for which an active warden is implemented that is able to work in real-time.

In 2000 a competition has been started on digital watermarking robustness, announced by the Secure Digital Music Initiative (SDMI). The goal was to break various audio watermarking techniques, but no restriction was given about the resulting quality [6]. In [44] attack approaches on SDMI watermarks have been published. While some of these attacks require the study of the embedding algorithm (*non-blind* attacks), they also mention pitch shifting and "time axis warping". But although it is said that their approaches successfully attacked the SDMI watermarks, no actual evaluation results of the strategies are given. Therefore, performed attacks may as well degrade the audio quality by an unacceptable amount while our techniques try to keep up the original quality as much as possible.

Related to our approach is the work from [3] which deals with attacks on digital audio watermarking systems. In this work watermarking attacks are grouped into four categories: Removal, desynchronization, embedding and detection. For each group, several attacks are described but no evaluation is done to show the actual efficiency of the attack categories as well as their processing time. Our approach enhances this work by giving an overview of the performance as well as the audibility of the prevention techniques.

The idea of steganography jamming by a method called "double-stegging" is introduced by [1]. In this scheme a second embedding process shall disturb the receiver, trying to extract the first embedding. However no further jamming methods are pointed out and no evaluation about the efficiency of the proposed techniques is given.

2.2. **Detection of Embedded Data.** In the research from [32], VoIP communications are the main focus. It is stated that steganalysis is a tough task for VoIP streams because of the high bandwidth available. Due to this fact a scheme is presented which analyzes steganography in VoIP using misuse patterns. Such patterns depict the application of steganography and can as well be used to find ways for the prevention of occurred embedding [32]. The system from [24] uses various information, statistically gathered and

extracted from the cover medium, in order to detect embedded messages. First the characteristic of the embedded message is used as input for the detection process. As different formats, e.g. plain text or encrypted data, produce different statistics, this is an important factor for steganography detection. Further the characteristics of the cover medium and the steganographic algorithm are considered as well. For example, the difference between the original and modified cover medium reveals various information about the embedding [24].

The main issue with these approaches is the fact that their focus lies on the detection of embedded data, i.e. steganalysis. Afterwards ways are searched to prevent an existing steganographic communication [24, 32]. As steganalysis is a complex task which depends on the embedding scheme as well as the cover medium, such processes cannot be done in real-time, hence making those systems inapplicable for real-time environments. Further steganographic algorithms can be made robust against steganalysis, e.g. using the evaluation scheme from [27]. This makes them theoretically undetectable by steganalysis systems. However, modifications applied to the communications channel can still harm the steganographic receiver. Hence steganographic communications can still be disturbed, even if they are steganalysis-proof.

2.3. **Recovery of Embedded Information.** An early work on watermark recovery was done by [22], focussing on digital images as cover media. In the process of watermark embedding, this system inserted "feature points" which shall support the extraction process in recovering the embedded watermark. Dealing with digital images, tests have been done with operations like scaling and rotating.

As cropping is a common attack which can also happen due to lost packages during a network communication, [15] proposed an algorithm for audio data which can recover an embedded message even if a cropping attack occurred. However no other attack scenarios are considered, leaving this algorithm vulnerable to prevention techniques.

Another audio watermarking algorithm was introduced by [29]. Comparable to the feature points from [22], "salient points" are spotted inside the audio signal. Such points are for example chosen on positions with rapid energy transition. This is due to the consideration that such areas shall not be disturbed too much inside an audio signal as this would otherwise degrade the resulting audio quality [29]. Hence these salient points are used to support the extraction process. Tests with StirMark for Audio still demonstrate a high BER, especially when doing frequency domain attacks and cropping.

Summarizing it can be said that recovery algorithms can protect the embedded data in supporting the extraction process after one of a predefined set of attacks has been performed. However such watermarking or steganographic algorithms are still potentially vulnerable to other kinds of attacks.

Our work distinguishes itself from other publications by the fact that no steganalysis is performed and our approach focuses on audio steganography rather than digital watermarking [3] or images as cover media [42] and is able to perform in real-time on unstructured rather than structured carriers [10]. Further our attack techniques are combinable while StirMark for Audio only allows for one attack at a time [8, 33] and we enhance the StirMark for Audio suite by novel attack methods. Basic signal processing operations are combined in a novel way with the goal to interfere with the steganographic receiver. There are three important features: First, our system is not focused on a specific steganographic algorithm, but can be utilized on any audio data, even if no embedding was performed. Second, the attack techniques do not degrade the quality of the cover audio signal, making it suitable for different areas of application. Third, as only basic

operations are performed our system works in real-time, making it applicable for VoIP streams.

3. **Attack Techniques.** In this section the signal processing operations making up our novel approach are pointed out and their usage for the prevention of steganographic communications is described. The presented techniques can be divided into two categories, comparable to the removal and desynchronization attack categories from [3]. On the one hand we describe and implement techniques which model the behaviour of some communications channel or network connection ("natural" modifications). On the other hand, there are mechanisms which perform modifications on the audio signal that would not happen due to outer influences ("malicious" modifications). The categories embedding and detection which were further described by [3] are not in our focus. This is because these approaches are potentially more time-consuming while we aim at real-time processing without trying to detect any embedded data.

All techniques work block based, which means that the audio signal is processed in blocks $b_1, \ldots, b_n$ of same length. For techniques which work directly on the audio data in time domain the modifications on one block b_i can be performed sample by sample. To be able to work in a transform domain as well our preprocessing stage has the ability to transform a block b_i into its corresponding block in frequency domain, B_i. This is done via the fast fourier transform (FFT) algorithm. After modifications the inverse FFT algorithm is used in the postprocessing stage, transforming the block B_i back into the time domain block b_i.

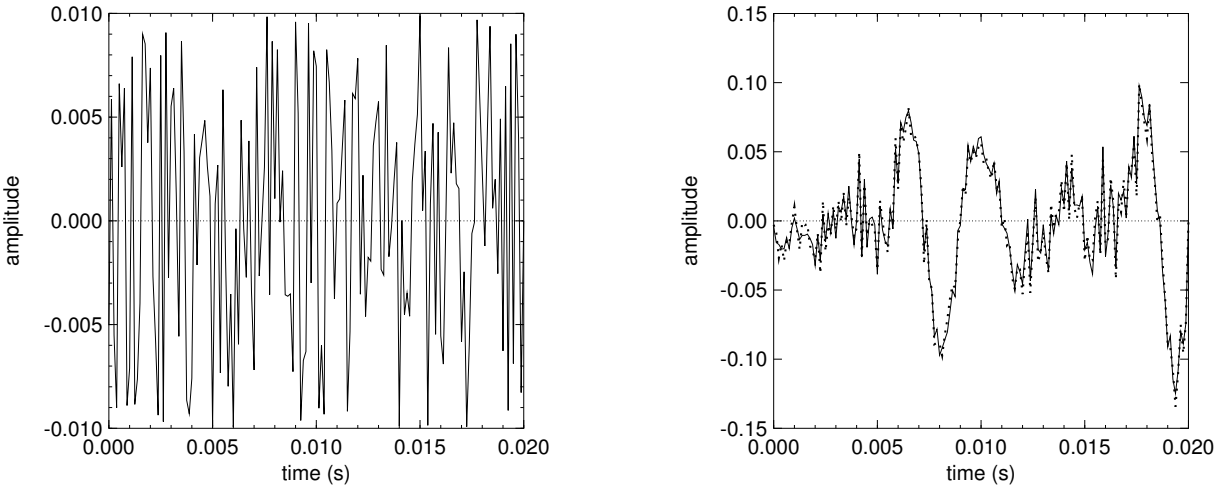
The following paragraphs describe the workings of each signal processing operation used in our approach on one block of the audio signal, b_i or B_i. Afterwards a note on the implementation is given, outlining important aspects of our implementation.

3.1. **"Natural" Modifications.** Techniques from this category model existing behaviours which have real-world causes. Belonging to this category are processes which have natural issues, like noise addition. Further, occurrences arising from network transmission behaviour and analogue transmissions are considered.

These effects can happen at any time, so an observer cannot tell whether the modifications have been done by our techniques or by some real-world event. As a consequence these techniques are not suspicious, even if they are detectable e.g. by analyzing the waveform of a signal.

White noise. An obvious technique for the disruption of steganography in any media is the addition of white noise, where individual samples are changed by a random value hence modeling electrical noise.

Noise addition or interference happens to transmitted signals on various communication channels, e.g. due to thermal noise or crosstalk. As a consequence audio data is affected as well when transmitted over such channels. Due to an increased noise level and changed sample values the steganographic receiver may not be able to extract all the embedded information correctly, depending on the actual embedding algorithm used. This operation directly influences the quality of the audio signal, which degrades with an increasing noise amplitude. In general a signal-to-noise ratio (SNR) above 20dB guarantees for a reasonable audio quality. As a consequence our approach uses very little noise amplitudes, therefore keeping the SNR above 20dB and minimizing the influence on the audio quality while still interfering with the steganographic receiver. Figure 1(a) shows an example of a noise signal which has been added to a block of the audio signal. The result shows Fig. 1(b), where the original signal is shown as solid curve and the audio signal including the noise is shown as dotted curve.



(a) White noise signal, zoomed (b) Original and modified signals

FIGURE 1. White noise addition

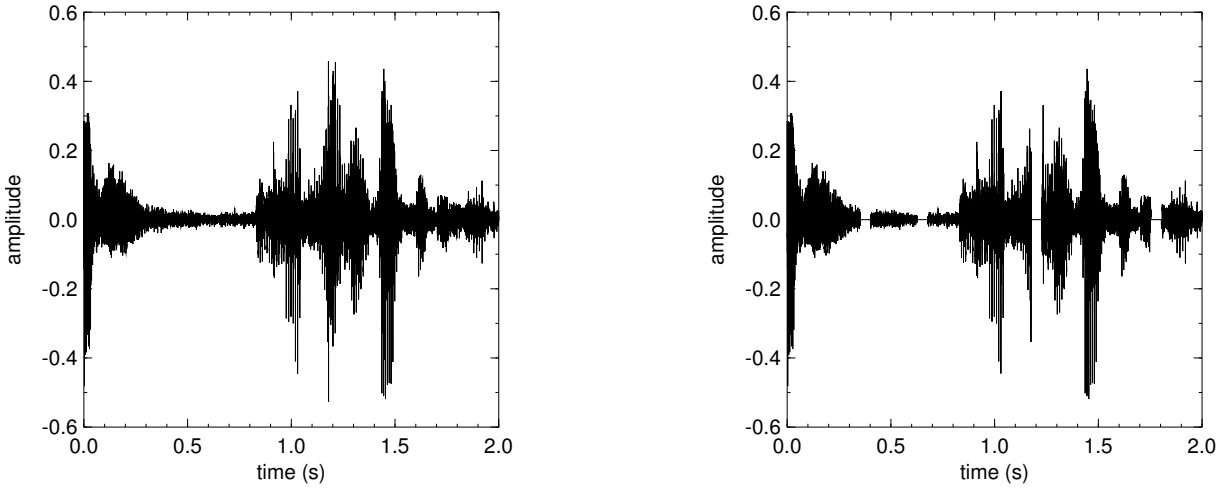
Packet loss. When dealing with VoIP transmissions the speech data is sent via RTP [37] which is built upon UDP, a connection-less data transmission protocol. In this context connection-less means that lost packets are not detected as there are no acknowledgements sent. This behaviour makes sense for telephony applications, as resent packets with speech data would make no sense at a later point of the conversation. Further the overhead of a connection-oriented protocol disappears.

However, for steganographic applications the loss of one speech packet is analogous to lost embedding information at the receiver. Depending on the area of application the lost time duration normally lies in the range of some milliseconds. For example VoIP transmits 20ms of speech within one RTP packet. As 20ms is such a short time duration, the fluency of the active conversation is not disturbed. From a steganographic point of view some data is eventually lost, depending on the steganographic algorithm and its actual extraction process.

As with other techniques the time duration for the packet loss modeling shall not be chosen above a certain threshold, otherwise the introduced pauses will wreck the quality of the ongoing conversation. In our prototype implementation at most 20ms of audio data, i.e. the content of one RTP packet, are discarded at once. Figure 2(a) shows 2s of an original audio signal while Fig. 2(b) shows the same signal after modifications. Blocks of 20ms have been discarded over the whole speech, where the gaps between those blocks were chosen randomly. The influences of this technique are noticeable but not suspicious for VoIP communications in congested networks.

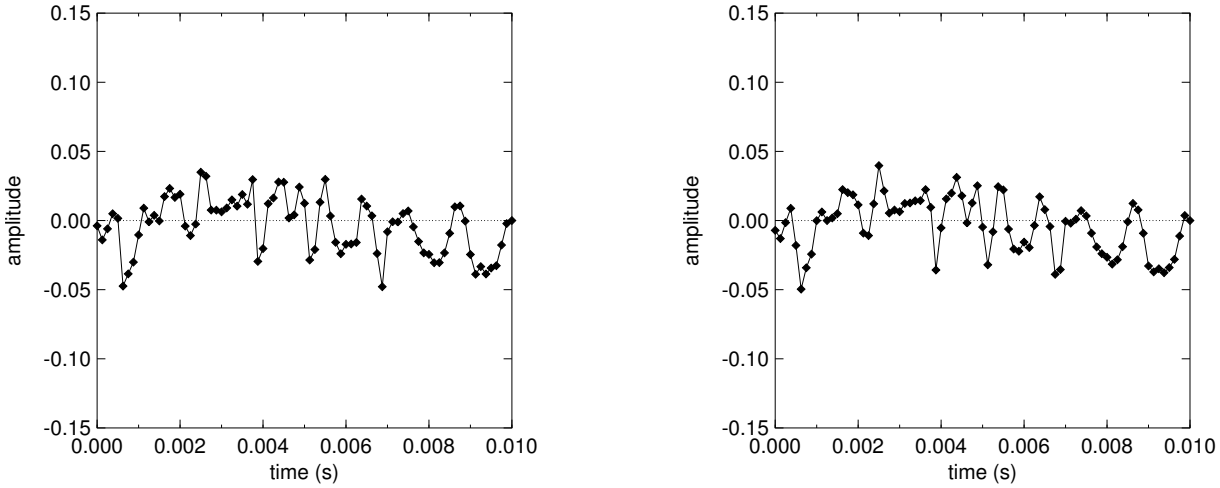
Sample shifting. This technique models the behaviour of resampling or an analog-to-digital (A/D) conversion. In the case of an over-the-air transmission the signal is re-sampled at the microphone of the receiver, hence the sampling points are shifted versus those at the sender. For steganographic algorithms such shifted sampling points may be a problem for the extraction process. This is especially true if no synchronization is done at the receiver. Therefore this technique can disrupt the steganographic communication while having no influence on the quality of the underlying audio signal.

Figure 3(a) shows an original audio signal, whereas Fig. 3(b) shows the same audio signal with samples shifted by the factor 0.38473 (48.09μs in this example). The sampling points on both plots have been highlighted for easier comparison.



(a) Original signal (b) Modified signal

FIGURE 2. Packet loss simulation



(a) Original signal (b) Modified signal

FIGURE 3. Sample shifting

3.2. **"Malicious" Modifications.** Techniques from this category perform malicious modifications. These are effects which would not occur in other circumstances, i.e. without the usage of our techniques. Due to this constraint these techniques have to take care that the modifications are not easily detectable by an observer. Further, an acceptable audio quality has to be preserved.

Variable time delay. If an audio signal contains static time delays the steganographic receiver can detect and bypass them. Therefore this approach deals with time delays whose duration varies over time, i.e. introducing delay jitter into the audio signal. This technique has already been briefly described in [35] but without giving a concrete implementation. Further the description from [35] only mentions a fixed block length where one sample is deleted or introduced, while our implementation is able to vary those parameters.

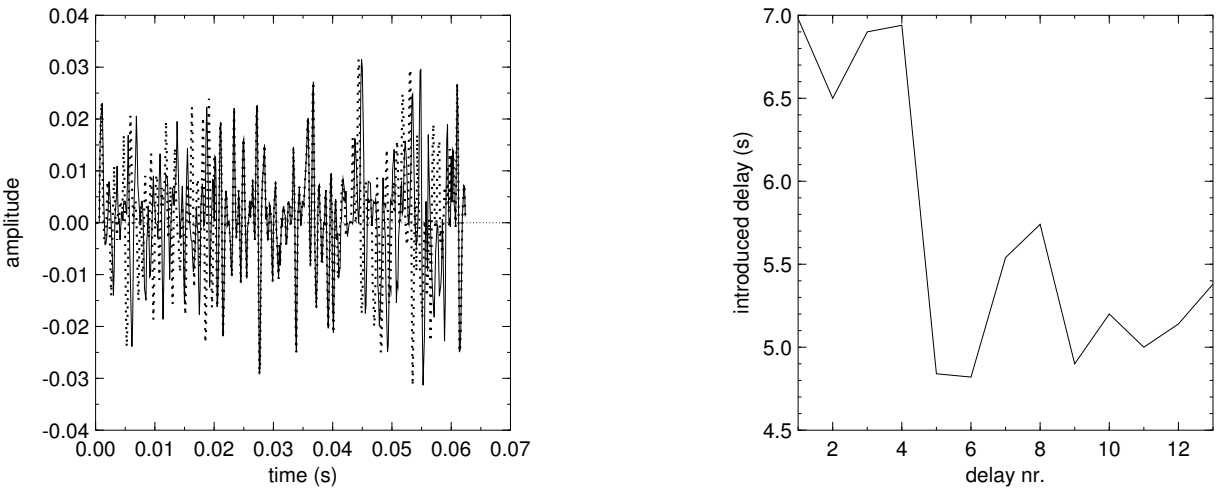
The advantage of this technique is that it does not influence the audio quality when a short delay time is chosen and the period, after which another delay is introduced, is large. Typical parameters for the tests of our prototype implementation introduced delays of less than 10ms over a period of about one second.

As such delays correspond to phase shifts in the frequency domain the receiver analyzes other sample values than the sender. Therefore steganographic algorithms which do not continuously synchronize themselves during extraction are seriously harmed by this kind of technique.

For the application of this technique two variables have to be defined. t_m is the time duration in which one sample is added or removed from the audio signal and t_d is the duration of the desired time delay. For both, t_m and t_d, only upper and lower limits are configured. The concrete values for the current block are chosen randomly at run time.

To introduce the variable delay these values are slightly modified for each new delay. Therefore the algorithm works in two steps: First, until reaching a delay of t_d one sample is added every t_m seconds. Second, after a delay of t_d seconds was introduced one sample is deleted every t_m seconds. Due to the second step the length of the audio signal nearly stays the same which is necessary for some environments like VoIP connections or when dealing with WAV files.

Figure 4(a) shows an audio signal without (solid curve) and with introduced variable delays (dotted curve). As it can be seen, after a delay of duration t_d is reached the signals start overlapping each other, meaning that the delay which was added beforehand is cleared. Afterwards another delay with a new duration t_d starts, deferring the dotted curve. An example of different delays generated during the processing of an example audio signal shows Fig. 4(b).



(a) Original and modified signals

(b) Delay variation over time

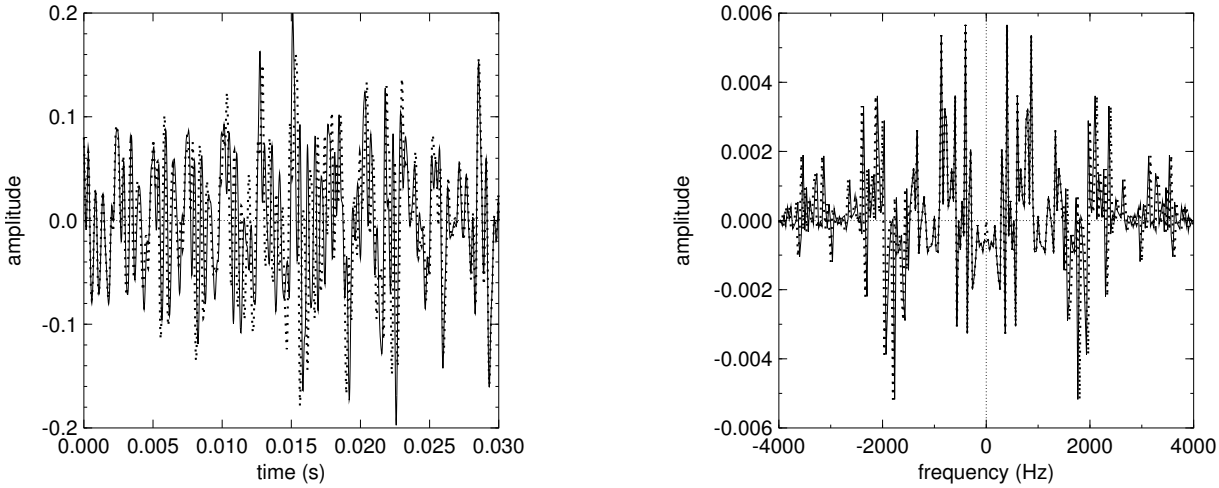
FIGURE 4. Variable time delay

Frequency shifting. This technique works in the frequency domain hence operating on blocks B_i. The goal is to shift the whole signal by a specific frequency factor Δf to disturb the steganographic channel while maintaining a good audio quality. The frequency factor Δf shall be a fraction of one halftone, i.e. fractions of $\sqrt[12]{2}$ as an octave consists of 12 halftones [17].

When changing the frequency of an audio signal it has to be noted that the individual frequencies must not be additively increased or decreased by Δf. This is due to the fact that frequencies and halftones are logarithmically related to each other [17]. As a consequence a multiplication by the halftone factor Δf is required for an undisturbed modification. When shifting all frequencies of an audio signal by this factor through multiplication, the resulting signal may sound clearer or darker. However the overall

quality stays the same. When choosing a small Δf (less than 1/5) no changes in the resulting sound are noticed (see Sect. 4).

Figure 5 shows 30ms of an example audio signal which was shifted by the factor 1.011546, about 19% of one halftone. In Fig. 5(a) the impacts on the time domain are presented (the original signal is shown as solid curve while the modified one is shown as dotted curve). As it can be seen the effects of this technique on the time domain signal are quite severe. However as mentioned before, if Δf is chosen appropriately no audible mutations are noticeable. In contrast Fig. 5(b) compares the original (solid curve) and modified (dotted curve) signals in the frequency domain, showing frequencies up to 4kHz.



(a) Time domain (b) Frequency domain

FIGURE 5. Frequency shifting, original and modified signals

3.3. **Implementation.** The attack techniques mentioned above have been integrated into our software framework for steganographic communications [30], which was originally used for embedding and extraction. The framework currently is able to work with PCM, A-law and μ-law encoded audio signals, coming from an RTP stream (VoIP communication), the sound card or WAV files. Due to its modularity we were able to integrate the attack techniques as "special" embedding algorithms which are fed with audio data from the framework and return modified audio data for output or transmission.

We now want to note some implementation details and important aspects:

1. White noise is generated with a random number generator, where the `rand()` function from the *libc* is used [25]. This function does not generate as random output as for example the kernel random number source device `/dev/random`. Therefore we also optionally support the usage of this device file. It is not used by default because it will block if not enough entropy is available [26] which is not tenable for a real-time application.
2. Some techniques require an interpolation of the processed signal for the following manipulation. For example when shifting samples we increase the sampling rate through interpolation to do the actual shifting by some fraction of the original audio sampling rate. For this task the GNU scientific library (*gsl*) [13] is used, in particular their implementation of the spline interpolation. In order not to be forced to do extrapolation we always keep one sampling point from the previous processed signal block and perform the interpolation between the kept sample and the last read sample.

3. As already mentioned our framework and hence also the built-in attack techniques are able to work in real-time. However if a conversion of a block of samples into the frequency domain is necessary, a short delay has to be introduced. In the case of our proposed techniques, the frequency shifting method obviously works in the frequency domain. Therefore, at the start of the communication it has to wait for a configurable amount of time (normally in milliseconds) to perform the first FFT. As a consequence a delay is introduced on the communications channel, corresponding to this time duration. By default the block length for one FFT is set to 10ms. Hence the implementation complies with the ITU-T recommendation for a maximum one-way delay of 150ms [21]. For FFT and IFFT operations the *fftw* library is used [9].

For the tests of our proposed techniques (see Sect. 4), three instances of our framework have been employed: One end performed the embedding, a "Man in the Middle" executed some kind of attack and the other end tried to extract secret data.

4. Evaluation.

This section shows the performance of our attack techniques. First the quality of audio files before and after modifications have been applied is tested in order to make a statement about the resulting quality difference due to our techniques. Afterwards the impacts of the individual attack methods on the steganographic receiver is demonstrated, showing bit error rates (BER) for three different embedding algorithms. Further the relationship between attack technique and embedding algorithm, based on the test results, is investigated and the possibility of error-correction codes (ECC) as a countermeasure against attack techniques is considered.

4.1. Audio Quality Evaluation.

To rate the quality of audio signals the Perceptual Evaluation of Speech Quality (PESQ) test [20] has been used. This test allows for an automatic rating of the audio quality by software, based on the experience of a human listening to a speech. PESQ grades the resulting quality according to the mean opinion score (MOS) scale [19] that was originally developed to rate the audio quality in subjective listening tests. This scheme uses a conversation opinion scale from 1 to 5 which is listed in Tab. 1.

For this test 9 audio signals have been composed, each containing 20s of a recorded telephone speech. On this signals all attack techniques were performed in a row and tested via the PESQ reference implementation (available at [18]). Table 2 shows the results and the actual configuration per attack technique, comparing the average MOS grade of the original signal to those of the signal after manipulations. For better expression of the influence on the audio quality by our techniques no data has been embedded into the audio signal beforehand. As it can be seen our modifications do not significantly degrade the audio quality. While signals ii and iii result in a MOS value around 3, those have been generated using "natural" techniques. That means even if the audio quality gets degraded, the occurrence of those disturbances is not suspicious as the modification may have as well been produced by a real-world event. The MOS grades of the "malicious" techniques are around 4, indicating a good quality preservation. Even the combination of two or more attack techniques still maintains the audio quality.

Comparison to StirMark for Audio. In this section we outline the MOS gradings for attacks available in StirMark for Audio. For a description of the individual attack types we refer to [8]. Table 3 shows the results for all StirMark for Audio attacks. It has to be noted that only the default configuration of StirMark for Audio has been used, therefore a higher MOS grade does not only mean a good audio quality preservation but also indicates that the embedded message may not have been destroyed. As it can be seen, even in the

Table 1. ITU-T conversation opinion scale for MOS [19]

Perceived distortion level	Quality	Grade
Imperceptible	Excellent	5
Perceptible but not annoying	Good	4
Slightly annoying	Fair	3
Annoying	Poor	2
Very annoying	Bad	1

Table 2. MOS grades for original and modified audio signals

#	Manipulation	Configuration	MOS grade
i	Original signal		4.500
ii	Noise addition	SNR: 25dB	3.183
iii	Packet loss	20ms every 200ms	3.308
iv	Sample shifting	Shifting factor: 20%	4.159
v	Variable time delay	Avg. jitter interval: 0.5s	4.238
vi	iv + v		4.204
vii	Frequency shifting	Semitone factor: 0.25	3.835
viii	v + vii		3.752
ix	iv + v + vii		3.753

default configuration some attacks severely degrade the resulting audio quality while our proposed techniques always aim at most minimal influence on the quality of the modified audio data. What Tab. 3 further shows is that the only mechanism available in both StirMark for Audio as well as our proposed attack techniques is the addition of white noise (*AddNoise*) which provides a slightly better MOS grade in our implementation.

The efficiency of StirMark for Audio attack types against our attack techniques is shown in the next section.

4.2. **Prevention Efficiency.** To grade the efficiency of the techniques the BER from the steganographic extraction process have been used. That is, the higher the BER the more effective the attack technique has been on the particular steganographic algorithm.

For all tests 80 bits have been embedded into a WAV file containing the recording of a telephone speech. For these tests all attack techniques have been integrated into a software framework which we originally implemented for steganographic communications over audio channels. Currently this framework supports three different steganographic algorithms: Echo hiding, spread spectrum in time domain and phase coding.

In the following the results of the steganographic receiver for each of those algorithms after application of our attack techniques are presented. Afterwards it is statistically tested whether the BER is related to the embedding algorithm as well as the chosen attack technique.

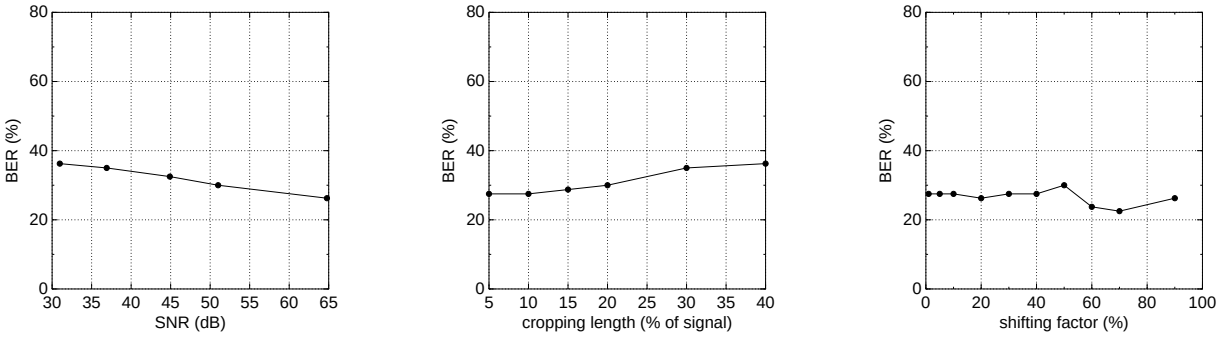
Echo hiding. This algorithm adds an echo on continuous blocks of the cover medium to embed a secret bit. The echo delay, decay as well as the block size are configurable [14].

Figures 6 and 7 show the results of the echo hiding receiver when our modifications are applied. As pointed out by Fig. 6(a), noise addition affects the echo hiding receiver when the resulting SNR gets below 50dB. Regarding packet loss the BER starts increasing when more than 10% of the signal length is lost (see Fig. 6(b)). Figure 6(c) depicts the last of our "natural" modifications, sample shifting, whose impact on the echo hiding receiver is not as high as with the other techniques. This is because only the position of an echo

TABLE 3. MOS grades for StirMark for Audio attack types

Attack type	MOS grade	Attack type	MOS grade
AddBrumm	4.500	FlippSample	2.287
AddDynNoise	3.839	Invert	4.282
AddFFTNoise	3.112	LSBZero	4.500
AddNoise	3.020	Noise_Max	3.079
AddSinus	4.477	Normalizer1	3.758
Amplify	4.500	Normalizer2	2.354
BassBoost	4.500	RC_HighPass	4.290
BitChanger	4.500	RC_LowPass	4.499
Compressor	3.653	ReplaceSamples	3.960
CopySample	2.516	Smooth2	3.285
CutSamples	3.746	Smooth	2.892
Echo	2.648	Stat1	4.436
Exchange	2.103	Stat2	4.496
ExtraStereo	2.820	VoiceRemove	2.366
FFT_HLPassQuick	3.473	ZeroCross	2.789
FFT_Invert	4.479	ZeroLength1	2.071
FFT_RealReverse	2.141	ZeroLength2	3.031
FFT_Stat1	2.141	ZeroRemove	3.858

is examined while the actual sample values do not influence this process. While [35] mentioned that echo hiding is robust against jitter attack, our implementation of variable time delay leads to BER around 30% (see Fig. 7(a)). This is due to the fact that we not only implemented a static jitter but rather our implementation varies the jitter time as well as the block length. Because little modifications in the frequency domain lead to a large number of modifications in time domain, there is a huge impact of frequency shifting (Fig. 7(b)) on the echo hiding receiver, hence giving a high BER.

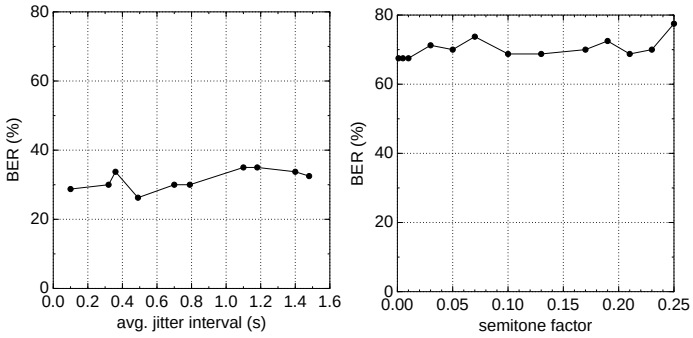


(a) Noise addition (b) Packet loss (c) Sample shifting

FIGURE 6. BER for echo hiding, "natural" modifications

Spread spectrum in time domain. The secret message is spread to a chip sequence to gain robustness. For embedding the chip sequence is modulated onto a sine carrier which is afterwards added to the cover audio signal.

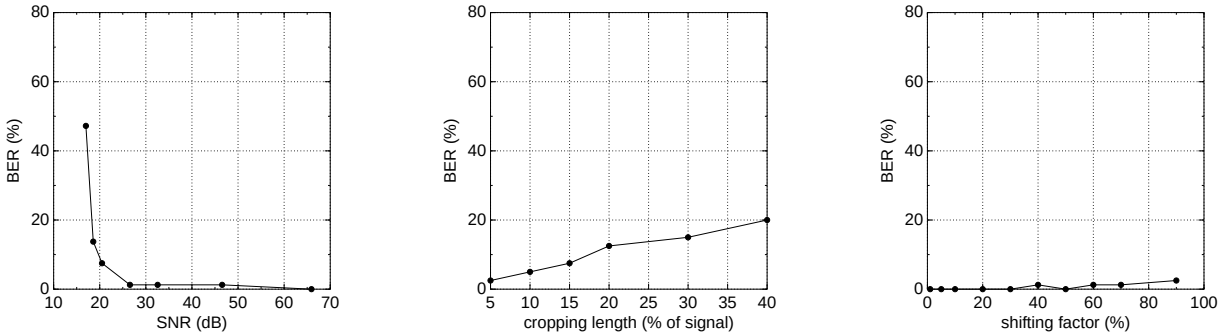
Figures 8 and 9 show the results of this algorithm with applied modifications. In Fig. 8(a) the robustness of the spread spectrum algorithm against noise addition becomes clear. The BER does not rise until the SNR falls below 18dB. Further, packet loss (see



(a) Variable time delay (b) Frequency shifting

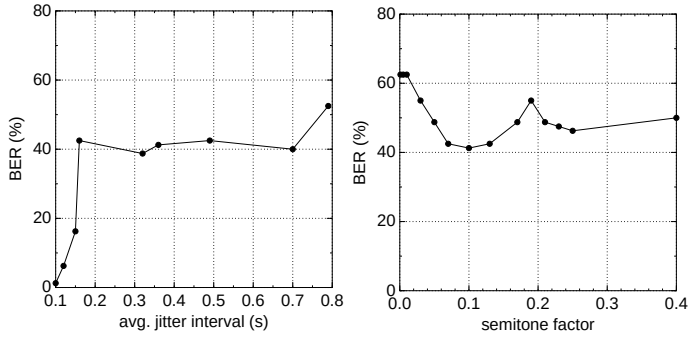
FIGURE 7. BER for echo hiding, "malicious" modifications

Fig. 8(b)) only increases the BER if more than 40% of the signal is lost. The modulated sine wave can still be extracted by the spread spectrum receiver if all sample values are shifted by a constant factor. Due to this fact Fig. 8(c) shows a BER around 0%. However if the delay starts to vary the receiver gets disturbed which results in high BER (see Fig. 9(a)). Also frequency shifting (Fig. 9(b)), which leads to large impacts in the time domain signal, severely influences the spread spectrum receiver.



(a) Noise addition (b) Packet loss (c) Sample shifting

FIGURE 8. BER for spread spectrum in time domain, "natural" modifications

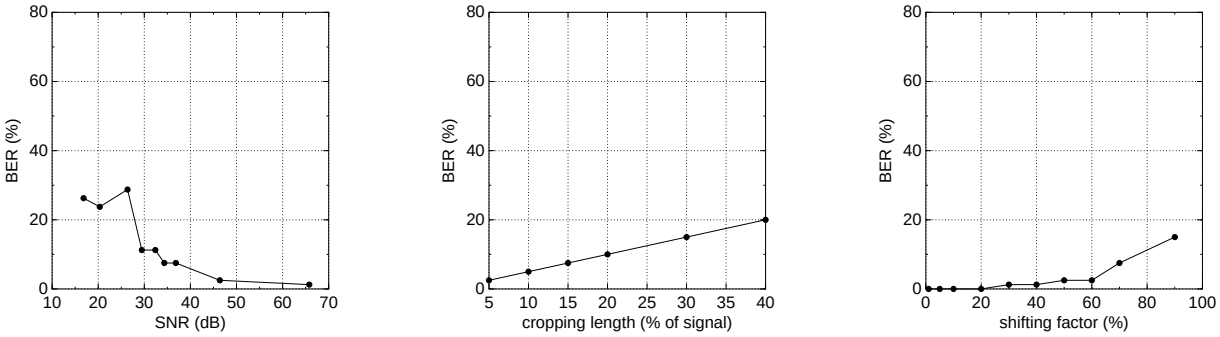


(a) Variable time delay (b) Frequency shifting

FIGURE 9. BER for spread spectrum in time domain, "malicious" modifications

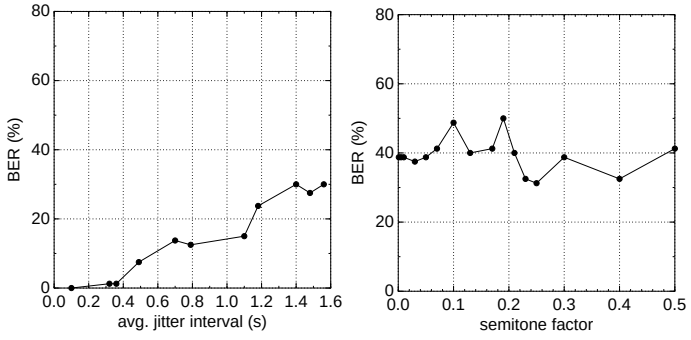
Phase coding. This algorithm works in the frequency domain. One bit is embedded by the introduction of a configurable mean phase difference between two adjacent parts of a configurable frequency interval for each block of the cover audio signal. For an enhanced robustness the bits are spread to a chip sequence [31].

Figures 10 and 11 show the BER when our modifications are applied to the phase coding algorithm. As with the previous algorithm, noise (Fig. 10(a)) starts influencing the receiver if the SNR falls below a certain value, which is about 30dB in this case. Also comparable to the previous outcomes is Fig. 10(b) which shows that BER and loss are proportional in relation to the length of the signal. Due to the fact that the shifting of samples values corresponds to a phase shift, the sample shifting technique has some impact on the phase coding receiver if the shift factor is above 70% (see Fig. 10(c)). Starting at an average delay variation of 1s Fig. 11(a) shows the severe impacts of variable time delay on the resulting BER. Furthermore, because this algorithm works in the frequency domain the frequency shifting technique also disturbs the receiver, giving a high BER (Fig. 11(b)).



(a) Noise addition (b) Packet loss (c) Sample shifting

FIGURE 10. BER for phase coding, "natural" modifications



(a) Variable time delay (b) Frequency shifting

FIGURE 11. BER for phase coding, "malicious" modifications

Relation between Embedding Algorithm and Attack Technique. After we showed the results of our attack techniques on the BER for different steganographic algorithms the relation between the actual algorithm as well as the used prevention mechanism shall be considered. For this statistical χ^2 test we took the average BER per algorithm and attack technique, as shown in Tab. 4.

The resulting χ^2 value is 31.57. Therefore, having 8 degrees of freedom and requiring a statistical significance of $(1 - \alpha) = 0.95$, we conclude a relationship between the applied

embedding algorithm in combination with a specific attack technique. This is in accordance with the above results, as for example the shifting of samples only affects the echo hiding algorithm while the other two algorithms produce an average BER of only 3% or less.

TABLE 4. Mean values of BER (in %) per algorithm and attack technique

	Embedding algorithm		
Attack technique	Echo hiding	Spread spectrum	Phase coding
Noise addition	32.0	10.32	13.33
Packet loss	30.83	10.42	10.0
Sample shifting	26.6	0.69	3.0
Variable time delay	31.5	31.25	14.77
Frequency shifting	70.28	50.98	39.375

Comparison to StirMark for Audio. While the previous section contrasted the audio quality preservation of StirMark for Audio with our approach, the efficiency in relation to our attack techniques are highlighted in this section. For the tests, all StirMark for Audio attacks have been applied to the same telephone speech recording containing 80 bits of embedded data that was also used to test our proposed techniques. As embedding algorithm, spread spectrum in time domain has been used.

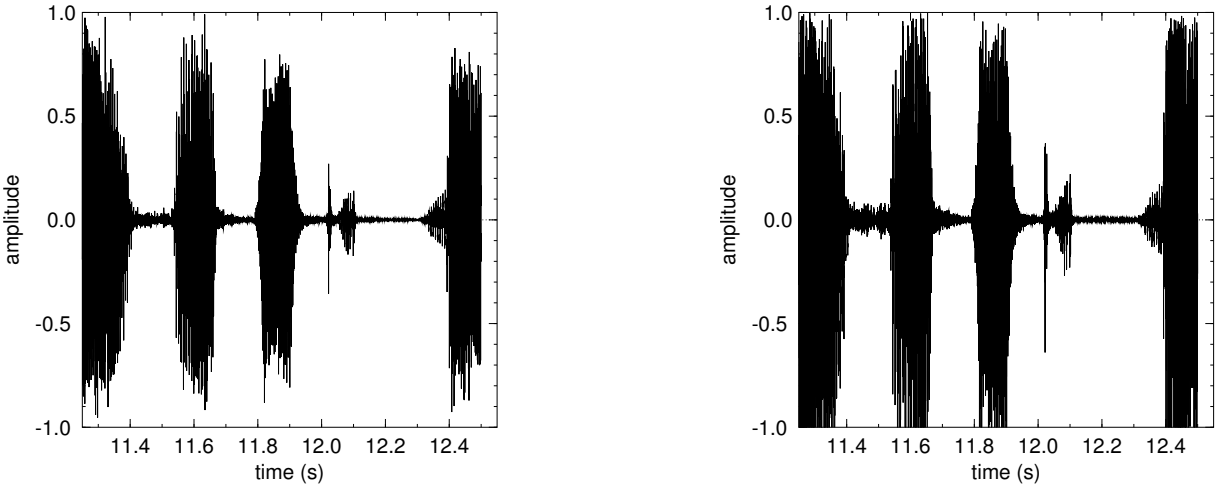
Table 5 shows the resulting BER for all StirMark for Audio attack types. Again the default configuration has been used. Only those attacks have been included in the table that produced a BER of more than 0%. Similar to our observations with frequency shifting, the frequency domain attack from StirMark for Audio (*FFT_HLPassQuick*) also leads to a high BER. Other attacks which disturb the spread spectrum receiver are those which flip samples (*CopySample*, *CutSamples*), comparable to our variable time delay method. While the *ZeroLength*{*1,2*} attacks also produce high BER they are also destructive to the resulting audio quality, as shown in Tab. 3. This assertion is also true for *ExtraStereo*. For example Fig. 12 compares the waveform of the audio signal before and after the *ExtraStereo* attack. As it can be seen in Fig. 12(b) the attack overdrives the signal amplitude, leading not only to high BER but especially to a poor audio quality.

Summarizing it can be said that not all attacks from StirMark for Audio are suitable for our setting, in interaction with the tested steganographic algorithms. Two important aspects distinguish our proposed methods from StirMark for Audio. First the only real overlap between StirMark for Audio attacks and our attack methods is the noise addition attack. All other methods that we implemented are novel and therefore do not exist in StirMark for Audio in this shape. Second, and very important as this was one of our main design goals, is the fact that while some StirMark for Audio attacks lead to a high BER, many of those attacks also degrade the resulting audio quality. This fact renders them only very limited usable for a scenario which we included as possible area of application: The (real-time) execution of attack methods on any link inside a network, independent of the fact whether steganography is really practiced. Such an application has a high demand on keeping the original audio quality, as non-steganographic communications do not want to be influenced by the application of such attack mechanisms.

4.3. **Error-correction Codes.** Having the BER shown in the previous section we now want to consider the application of ECC as a countermeasure against steganography prevention. While various kinds of ECC do exist [28], we limit this discussion to the popular Hamming code.

TABLE 5. BER (in %) for StirMark for Audio attack types

Attack type	BER	Attack type	BER
CopySample	46.875	Normalizer2	2.5
CutSamples	56.25	Smooth	1.25
Exchange	1.25	Stat1	12.5
ExtraStereo	68.06	ZeroCross	25.0
FFT_HLPassQuick	63.75	ZeroLength1	48.75
FFT_Invert	2.5	ZeroLength2	48.75
FlippSample	6.25	ZeroRemove	48.61
Invert	1.25		



(a) Original signal (b) Modified signal

FIGURE 12. StirMark for Audio *ExtraStereo* attack

The Hamming code uses a block length of N bits, having $k = N - n$ parity bits and n data bits. It has the ability to correct one wrong bit in n transmitted bits [16]. As for the tests in Sect. 4.2 80 bits have been embedded into the cover audio data, we assume a block length $N \leq 80$. As $N = 2^k - 1 \leq 80 \Rightarrow k \leq 6$, we choose $k = 6$ parity bits and therefore are able to transmit $n = 2^k - k - 1 = 57$ data bits, using a block length of $N = 63$.

Table 6 shows the mean BER per attack technique, averaged over the three tested algorithms as well as the resulting number of incorrectly extracted bits from one block of 63 bits. As it can be seen from Tab. 6 any of our attack techniques disturbs the embedded data in a way that a $(63, 57)$ Hamming code would not be able to correctly restore the original message.

What can also be seen from Tab. 6 is that the "malicious" techniques result in higher average BER than the "natural" modifications, showing the shifting of the audio frequency by some amount of a semitone as the most effective disturbance method.

5. **Outlook and Future Work.** This section gives ideas for future research and use of the proposed attack techniques.

Using our software framework [30] we demonstrated the applicability of the proposed approach for steganography prevention. However the results of a specific attack technique depend not only on its method, but also on the algorithm used to embed data into the cover medium in the first place. The goal of this paper was to demonstrate the applicability

TABLE 6. Mean values of BER (in %) per attack technique and incorrect bits ($N = 63$)

"Natural" modifications			"Malicious modifications"	
Noise addition	Packet loss	Sample shifting	Variable time delay	Frequency shifting
18.55	17.08	10.09	25.84	53.545
11.69 bits	10.75 bits	6.36 bits	16.28 bits	33.72 bits

of our attack techniques in combination with basic steganographic algorithms already implemented in our software framework. Future work will expand the evaluation of our approach to other, modern, steganographic techniques like [45] embedding into wavetables and [43] dealing with MP3 coded audio signals. The implementation will be done into our existing software framework as we already integrated the attack techniques therein and it provides all necessary components for the embedding and extraction processes for audio steganography.

Besides the extension of tests to other steganographic algorithms future research will investigate the usefulness of further attack techniques, e.g. adapted to specific scenarios. Further a study on the combination of various attack techniques is planned, examining advantageous and disadvantageous combinations in general and for concrete scenarios.

6. **Conclusion.** In this paper several attacks on steganography with audio data as cover medium are given. The proposed techniques can be divided into two categories. The first category contains methods which perform "natural" modifications, i.e. their impact is modeled after real-world causes. Techniques from this category perform noise addition or the shifting of sampling values and simulate packet loss. Such modifications can happen at any time. As a consequence they do not look suspicious for an observer. Methods from the second category perform "malicious" modifications, i.e. such effects would not happen due to other causes than using our approach. Examples from this category are the introduction of variable time delay into the whole audio signal and the shifting of the audio frequency based on semitones. All techniques aim at maintaining an acceptable audio quality.

For the execution of these attack techniques only basic signal processing operations are performed. Therefore our proposed approach is easily usable and can be applied to real-time environments such as VoIP communications.

A prototype has been implemented into our software framework, originally used for steganographic communication over audio channels. For tests of the prevention efficiency the steganographic algorithms available in the software framework have been used. Efficiency tests yielded good performance of the attack techniques on different steganographic algorithms. Further the qualities of the resulting audio signals have been tested, approving the claimed preservation of the audio quality. A comparison with the StirMark for Audio benchmark further showed that our approaches do have novel elements and while StirMark for Audio also has successful attacks, it does not control the audio quality preservation in general, which was one of our main design goals. These results demonstrate the significance of our implementation besides StirMark for Audio. Therefore future research on this topic and the expansion of tests to other steganographic algorithms is justified.

Acknowledgment. Our research project is funded by the KIRAS PL3 program of the Austrian Research Promotion Agency (FFG).

The author wants to thank the reviewers for their helpful comments and suggestions, which have improved the presentation.

REFERENCES

[1] Sally Adee, Spy vs. spy., http://spectrum.ieee.org/computing/software/spy-vs-spy/.

[2] Z. K. Al-Ani, A. A. Zaidan, B. B. Zaidan and H. O. Alanazi, Overview: Main fundamentals for steganography, *Journal of Computer*, vol. 2, no. 3, pp. 158-165, 2010.

[3] M. Arnold, Attacks on digital audio watermarks and countermeasures, *Proc. of International Conference on Web Deliv. Music*, pp. 55-62, 2003.

[4] P. N. Basu and T. Bhowmik, On embedding of text in audio-a case of steganography, *Proc. of International Conference on Recent Trends in Information, Telecommunication and Computing*, pp. 203-206, 2010.

[5] W. Bender, D. Gruhl, N. Morimoto and A. Lu, Techniques for data hiding, *IBM Systems Journal*, vol. 35, no. 3, pp. 313-336, 1996.

[6] S. A. Craver, M. Wu, B. Liu, A. Stubblefield, B. Swartzlander, D. S. Wallach, D. Dean and E. W. Felten, Reading between the lines: lessons from the SDMI challenge, *Proc. of the 10th USENIX Security Symposium*, 2001.

[7] J. Dittmann, A. Lang and M. Steinebach, Stirmark benchmark: audio watermarking attacks based on lossy compression, *Proc. of SPIE*, pp. 79-90, 2002.

[8] J. Dittmann, M. Steinebach, A. Lang and S. Zmudizinski, Advanced audio watermarking benchmarking, *Proc. of SPIE*, pp. 224-235, 2004.

[9] FFTW, Fastest fourier transform in the west, http://www.fftw.org.

[10] G. Fisk, M. Fisk, C. Papadopoulos and J. Neil, Eliminating steganography in internet traffic with active wardens, *Proc. of the 5th International Workshop on Information Hiding*, pp. 18-35, 2002.

[11] J. J. Fridrich, M. Goljan, D. Hogea and D. Soukal, Quantitative steganalysis of digital images: estimating the secret message length, *ACM Multimedia Systems Journal*, vol. 9, no. 3, pp. 288-302, 2003.

[12] J. J. Fridrich, M. Goljan and D. Soukal, Searching for the stego-key, *Proc. of SPIE*, pp. 70-82, 2004.

[13] GNU, GSL gnu scientific library, http://www.gnu.org/software/gsl/.

[14] D. Gruhl, A. Lu and W. Bender, Echo hiding, *Proc. of the 1st International Workshop on Information Hiding*, pp. 293-315, 1996.

[15] A. Gurijala, Robust algorithm for watermark recovery from cropped speech. *Proc. of International Conference on Acoustics, Speech, Signal Processing*, vol. 3 , pp. 1357-1360, 2001.

[16] R. W. Hamming, Error detection and error correction codes, *Bell System Technical Journal*, vol. 2, pp. 147-160, 1950.

[17] D. Howard and J. Angus, /em Acoustics and Psychoacoustics, Focal Press, 4th edition, Burlington, USA, 2009.

[18] ITU-T, PESQ reference implementation, http://www.itu.int/rec/dologin pub.asp?lang=e&id=T-REC-P.862-200102-I!!SOFT-ZST-E&type=items.

[19] Recommendation ITU-T P. 800, Methods for subjective determination of transmission quality, *Technical Report*, 1996.

[20] Recommendation ITU-T P. 862, Perceptual evaluation of speech quality: An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, *Technical Report*, 2001.

[21] Recommendation ITU-T G. 114, One-way transmission time, *Technical Report*, 2003.

[22] N. Johnson, Z. Duric and S. Jajodia, Recovery of watermarks from distorted images, *Proc. of the 3rd Workshop Information Hiding*, pp. 318-332, 2000.

[23] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, Boston, London, pp. 121-148, 2000.

[24] R. Lang and H. Lu, Algorithm for detecting steganographic information based on characteristic of embedded message, *Proc. of International Conference on Intelligent Information Hiding Multimedia Signal Processing*, pp. 64-67, 2009.

[25] Linux, Man page-rand(3), http://linux.die.net/man/3/rand.

[26] Linux, Man page-random(4), http://linux.die.net/man/4/random.

[27] G. Luo, X. M. Sun, L. Y. Xiang and J. W. Huang, An evaluation scheme for steganalysis-proof ability of steganographic algorithms, *Proc. of International Conference on Intelligent Information Hiding Multimedia Signal Processing*, vol. 2 pp. 126-129, 2007.

[28] D. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, England, 2004.

[29] H. Malik, R. Ansari, and A. Khokhar, Robust audio watermarking using frequency-selective spread spectrum, *Proc. of IET Information Security*, vol. 2, no. 4, pp. 129-150, 2008.
[30] M. Nutzinger and R. Poisel, Software architecture for real-time steganography in auditive media. *Proc. of International Conference on Computing Technology Electronic Engineering*, pp. 100-105, 2010.
[31] M. Nutzinger and J. Wurzer, A novel phase coding technique for steganography in auditive media. *Proc. of the 6th International Conference on Available, Reliable, Secure*, pp. 91-98, 2011.
[32] J. C. Pelaez, Using misuse patterns for voip steganalysis, *Proc. of International Workshop on Database Expert Systems Application*, pp. 160-164, 2009.
[33] F. A. P. Petitcolas, StirMark Benchmark 4.0. http://www.petitcolas.net/fabien/watermarking/stirmark/.
[34] F. A. P. Petitcolas, Watermarking schemes evaluation, *IEEE Signal Processing*, vol. 17, no. 5, pp. 58-64, 2000.
[35] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, Attacks on copyright marking systems, *Proc. of the 2nd International Workshop Information Hiding*, pp. 219-239 , 1998.
[36] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, Information hiding-a survey, *Proceedings of IEEE*, vol. 87 , pp. 1062-1078, 1999.
[37] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications, RFC 355*, 2003.
[38] L. Shelley and J. Picarelli, Methods not motives: Implications of the convergence of international organized crime and terrorism, *International Journal of Police Practice and Research*, vol. 3, pp. 305-318, 2002.
[39] SourceForge, StirMark for audio, http://sourceforge.net/projects/stirmark/.
[40] M. Steinebach, F. A. P. Petitcolas, F. Raynal, J. Dittmann, C. Fontaine, C. Seibel, N. Fates and L. C. Ferri, Stirmark benchmark: audio watermarking attacks, *Proc. of International Conference on Information Technology: Coding Computing*, pp. 49-54, 2001.
[41] H. Wang and S. Wang, Cyber warfare: steganography vs. steganalysis, *Communications of the ACM*, vol. 47, no. 10, pp. 76- 82, 2004.
[42] A. Westfeld and A. Pfitzmann, Attacks on steganographic systems, *Proc. of the 3rd International Workshop on Information Hiding*, pp. 61-76, 1999.
[43] H. Wey, A. Ito, T. Okamoto and Y. Suzuki, Multiple description coding using time domain division for mp3 coded sound signals, *Journal of Information Hiding Multimedia Signal Processing*, vol. 1, no. 4, pp. 269-285, 2010.
[44] M. Wu, S. Craver, E. Felten, and B. Liu, Analysis of attacks on sdmi audio watermarks, *Proc. of International Conference on Acoustics, Speech, Signal Processing*, vol. 3, pp. 1369-1372, 2001.
[45] K. Yamamoto and M. Iwakiri, Real-time audio watermarking based on characteristics of pcm in digital instrument, *Journal Information Hiding Multimedia Signal Processing*, vol. 1, no. 2, pp. 59-71, 2010.