REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
31-03-2006	Final Report	1 May 2004 - 01-Nov-05

4. TITLE AND SUBTITLE

Steganalysis for Audio Data

5a. CONTRACT NUMBER
FA8655-04-1-3010

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Professor Jana Dittmann

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Otto-von-Guericke University of Magdeburg
Universitätsplatz 2
Magdeburg 39106
Germany

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 821 BOX 14
FPO 09421-0014

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Grant 04-3010

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Audio WET system is a web-based evaluation system which provides the user the functionality of different watermarking algorithms (embedding and detecting) with a large database of audio signals (test material). The system also provides both single and profile attacks which can be used to evaluate the watermarking algorithms. The user has the choice to specify the watermarking algorithms with their parameters for embedding and the type of audio content. Furthermore, the user selects the properties of the watermarking algorithms to be evaluated before starting the evaluation process.

15. SUBJECT TERMS
EOARD, Steganography, IP Telephony, Steganalysis

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			PAUL LOSIEWICZ, Ph. D.
UNCLAS	UNCLAS	UNCLAS	UL	38	19b. TELEPHONE NUMBER *(Include area code)* +44 20 7514 4474

Final Report – 2005, M 16 (original M12)

Otto-von-Guericke University Magdeburg, Germany
Working group: Multimedia and Security

Prof. Jana Dittmann, Andreas Lang, Claus Vielhauer
Thomas Vogel, Sandra Gebbensleben, Tobias Scheidat, Christian Krätzer

Steganalysis for Audio Data

a) Final report on Implementation and Evaluation of Digital Audio Watermarking Algorithms and Design of Distributed Parallel Attacks (Item 0004 / 0005)
b) Final Report on Demonstrator Software and Evaluation for speech steganography and steganalysis (Item 0004 / 0005)

1 Cover Sheet

Principal Investigator's name: Prof. Jana Dittmann

Institution's name: Otto-von-Guericke University of Magdeburg

Institution's address: ITI Research Group on Multimedia and Security, PO Box 4120, 39016 Magdeburg, Germany

Grant number: FA8655-04-1-3010

This final report summarizes the activity of (a) Benchmarking of Audio Data - The evaluation of audio watermarking algorithms using the Audio Watermark Evaluation and Test (WET) system, including test set development, test environment and test results; and (b) Steganography and Steganalysis for Audio Data - The Voice over Internet Protocol (VoIP) data embedding and hidden data detection scenario including, test set development, test environment and test results. The test results presented show for (a) show the evaluation of robustness, transparency, capacity and complexity of selected digital audio watermarking algorithms, and for (b) show the transparency and embedding capacity for VoIP steganography as well as the steganalysis of the transmitted data.

2 Objectives and Organizational Points

This final report summarizes the activities of (a) Benchmarking for Audio Data and (b) Steganography and Steganalysis for Audio Data, two concurrent tasks under the effort "Steganalysis of Audio Data". This project was sponsored by the European Office of Aerospace Research and Development (EOARD) and funded by the Air Force Research Laboratory, Multi-Sensor Exploitation Branch (AFRL/IFEC).

For task (a), the current features of the Audio Watermark Evaluation and Test (WET) system are presented. In addition the test environment, test set and results obtained through evaluation of five distinct watermarking algorithms, as well as the design developed for distributed parallel attacks, is introduced (see publication [LD06]). For task (b), the current

research results of the Voice over Internet Protocol (VoIP) scenario (VoIP scenario A with active steganography, [DHH05]) are presented, including test environment, test set and test results (see publication [VDHK06]).

3 Status effort

In this section introduces the following aspects of the Audio WET system:

(a.1) The current status of Audio WET implementation
(a.2) The design of distributed parallel attacking
(a.3) The test set used to evaluate Audio WET and parallel attacking framework
(a.4) The test environment employed
(a.5) The test results achieved

A very short introduction of Audio WET is also presented in Section a.1.

In section (b) the current status of the VoIP steganography and steganalysis task is discussed

(b.1) The test set is introduced
(b.2) The test environment employed
(b.3) The test results for the active steganography scenario.

(a.1) Current status of Audio WET

The Audio WET system is a web-based evaluation system which provides the user the functionality of different watermarking algorithms (embedding and detecting) with a large database of audio signals (test material). The system also provides both single and profile attacks which can be used to evaluate the watermarking algorithms. The user has the choice to specify the watermarking algorithms with their parameters for embedding and the type of audio content. Furthermore, the user selects the properties of the watermarking algorithms to be evaluated before starting the evaluation process.
Currently, the Audio WET system provides the following features:

- Menu-based web navigation
- Test data (The overall test set contains 1,072 audio files in a data base. For evaluation tests 398 of these were selected. See section a.3.1 for more information.)
- Functions:
  - Embedding with five different watermarking algorithms (See section a.3.2)
  - Retrieving the watermark information
  - Single attacks (40 provided by StirMark for Audio [LDSV05]) and tuned attacks, which have an improved transparency by employing psychoacoustic models [DKL05]. This development was supported with NoE Ecrypt project results [EC05].
  - Transparency measurement by computing ODG values in the interactive mode
  - Profile attacks using Embedding, Attacking, and Detection Profiles ([LD06], [LDLD05], [VD05], [LD04])
  - History of functions employed, including an option called “back tracking” to undo the last used function
- Visualization of audio signals (time, frequency and phase presentation)

A screen shot of the current audio WET system shown in Annex A, Section (a.1), Figure A1.

(a.2) Distributed parallel attacking

From the proposal, task (D) was undertaken to manage time consumption by evaluating many different watermarking algorithms with a large set of audio data. In order to accomplish this task effectively the design and implementation of parallel attacking was useful in reducing the test duration. Performance enhancements are one major challenge and are analyzed to design a distributed attack by parallel computing.
In this subsection, the principle of enhancing the Audio WET system with distributed parallel embedding, attacking and retrieving processes is discussed.

In general, the Audio WET system gains a distributed processing capability (distributed parallel attacks) through XML-based communication, which is used to send the watermark evaluation steps to many batch processes on different nodes. To provide parallel embedding, attacking and retrieving processes, it is important that the attack process only be run when the embedding process has completed its work; the same is true for the retrieval process. For this the reason, the concept of parallel watermark attacking that was chosen for this project is one where the distribution is performed on the data rather than on the processing. Therefore, it is only applicable when more than one audio file is used (i.e. the batch mode). Each node computes the entire task processes, but for only one audio file. Another audio file is used for embedding, attacking and retrieving by another node.

A task server *W* (the Audio WET system) is connected to a number of nodes *n*, denoted (N_0, N_1 ... N_{n-1}). *W* will then send the task *T* (represented in a XML structure) to all nodes together with a node-count *NC*. Each of the nodes takes its part of *T* and processes the data, with the result of the processing communicated back to *W*. Figure 1 shows the general task principle for a setup with three nodes.

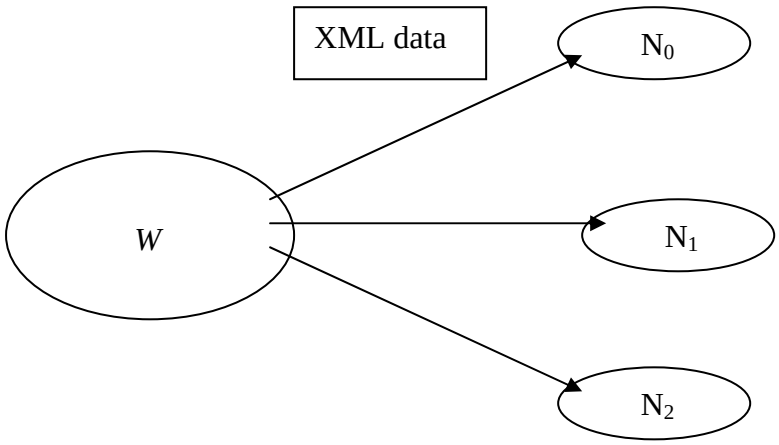

Figure 1: Task distribution to three clients

This batch processing model can be divided into the following phases:

1. Preparation of the XML structure containing the task *T* by *W*
2. Sending of the XML structure and the node count *NC* by *W*
3. Each of the *n* nodes identifies from *T* the share of data it has to compute

4. Each of the *n* nodes performs the computations necessary on the data
5. Each node sends its results back to *W*

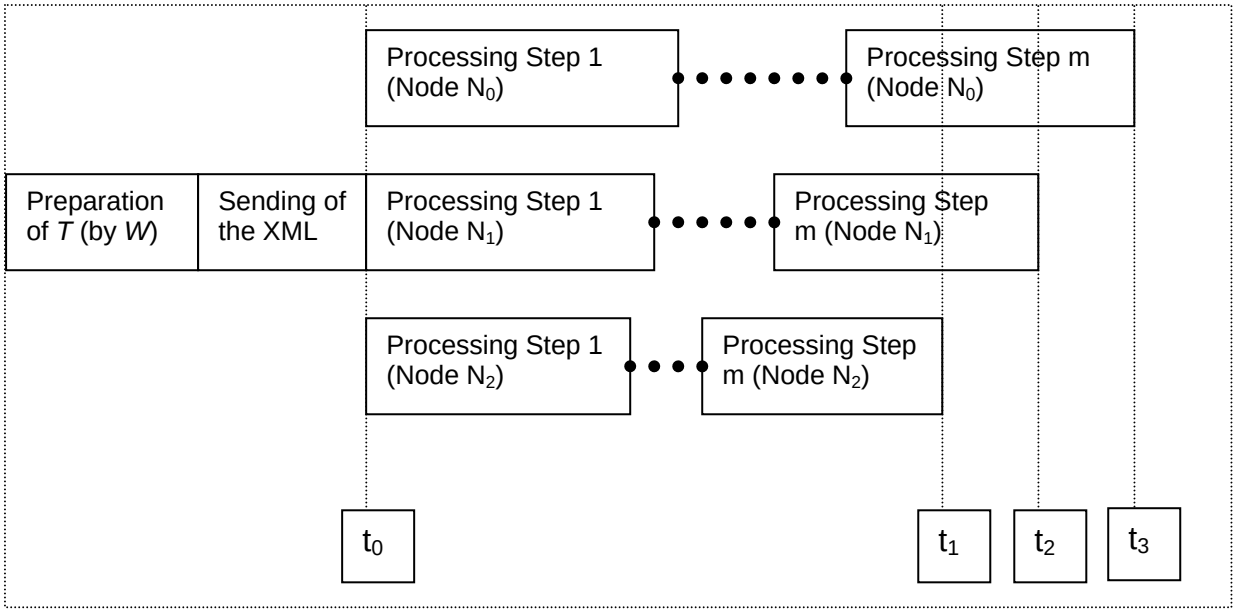

Figure 2: Distributed parallel attacks for three nodes with different computation times

Depending on the computation power of each individual node N_0 to N_{n-1}, the computations may be finished at different times (Figure 2). Figure 2 depicts the processing of *T* by three nodes, with the processing on the nodes finishing at different times (t_1, t_2 and t_3) despite starting at the same point of time (t_0). The entire task must be completed before a new task can be run.

The XML structure describes the task *T* to be performed by *W*. *T* consists of embedding, retrieval and attacking steps which are executed in sequence for each audio file provided.
The computation of the work sequence could be improved by distributing the steps to all known nodes. For this project, however, it was decided to base distribution on the files to be processed in order to minimize the bandwidth needed and ensure the correct sequence of embedding, attacking and retrieving the watermarking information for each selected audio file. Due to this decision, the XML structure must therefore contain the filename and all processing instructions (together with the parameters necessary) to complete the task. Furthermore, the node N_i which is to perform modifications on the specified file F_i must obviously have access to that file.

(a.3) Test set used for the tests

This subsection introduces the audio signals which are available for used in the evaluation process. Following this, audio watermarking algorithms used and parameters chosen are examined.

(a.3.1) Audio test set

The audio test set provided by the Audio WET data base contains 1,072 different audio files. For the evaluation tests a selection of this set is used which provides an equal distribution of

different types of audio content. This set consists of 389 selected audio files, which are divided into the four main categories identified below. Of these, some files were obtained royalty–free while others are ripped from original audio CD's. All audio files are PCM coded WAVE files with 44100 Hz sampling rate, 16 bit quantization and 2 channels (stereo), equating to standard audio CD format. Each file has a duration of about 30 seconds.
These main categories are:

- Music
- Sounds
- Speech
- SQAM (Sound Quality Assessment Material)

The breakdown for each of these categories is as follows:

- In the category Music a total of 267 files exist, which are distributed to ten sub-categories: *metal, pop, reggae, blues, jazz, techno, hip hop, country, classical* and *synthetic*. Additionally, the sub-category *classical,* with an additional 87 audio files, is again sub-divided into *choir, string quartet, orchestra, single instruments* and *opera*. The category *choir* contains 8, *string quartet* 18, *orchestra* 21, *single instrument* 20 and *opera* 20 audio files.
- The main category *sounds* is broken into four sub-categories (*computer generated, natural, silence* and *noise)* and contains 33 audio files. In *computer generated* are 12, *natural* 8, *silence* 2 and *noise* 11 audio files.
- The main category *speech* has four sub-categories (*male, female, computer generated* and *sports*). These sub-categories contains *male* 24, for *female* 20, for *computer generated* 21 for *sports* 11 audio files.
- The main category SQAM is a well known set used extensively for testing and consists of entirely royalty free files. There are 16 audio files, 9 voice and 7 instrumental files in this sub-category [SQAM].

For a better overview of all categories and sub-categories, the reader is referred to Figure A2 in Annex (a.1), which visualizes the audio test set.
During the evaluation it was discovered that one of the files contained in the test set is degenerated (it does not meet the file specifications of the test set and its data is not able to be processed by the evaluation suite). This fact did affect the measures slightly and its effects will be discussed in the appropriate places in the test results (Section a.5).

(a.3.2) Watermarking algorithms

For the evaluation of digital audio watermarking algorithms, five different algorithms were employed: Least Significant Bit (LSB), Spread Spectrum, Publimark, and two wavelet based algorithms – the Viper Audio Water Wavelet and AMSL Audio Water Wavelet algorithms. The following sub sections describe each algorithm and its applicable parameters.

(a.3.2.1) Least Significant Bit (LSB)

This blind watermarking algorithm embeds the watermark message into the Least Significant Bits (LSB) of the audio signal and is described in [King99]. The algorithm has two general implemented modes: with and without the secret key (*k*). If no *k* is used, the message is embedded into all LSBs of the audio signal. If *k* is used, then the watermark is not embedded in all LSBs. This is due to the fact that the key initializes a Pseudo-Random Number Generator (PRNG), whose values are used to scramble the embedding position and to select the used LBSs. Therefore not all LSBs are used, decreasing the capacity and increasing the transparency. The following Figure 3 introduces the scramble scheme in detail.

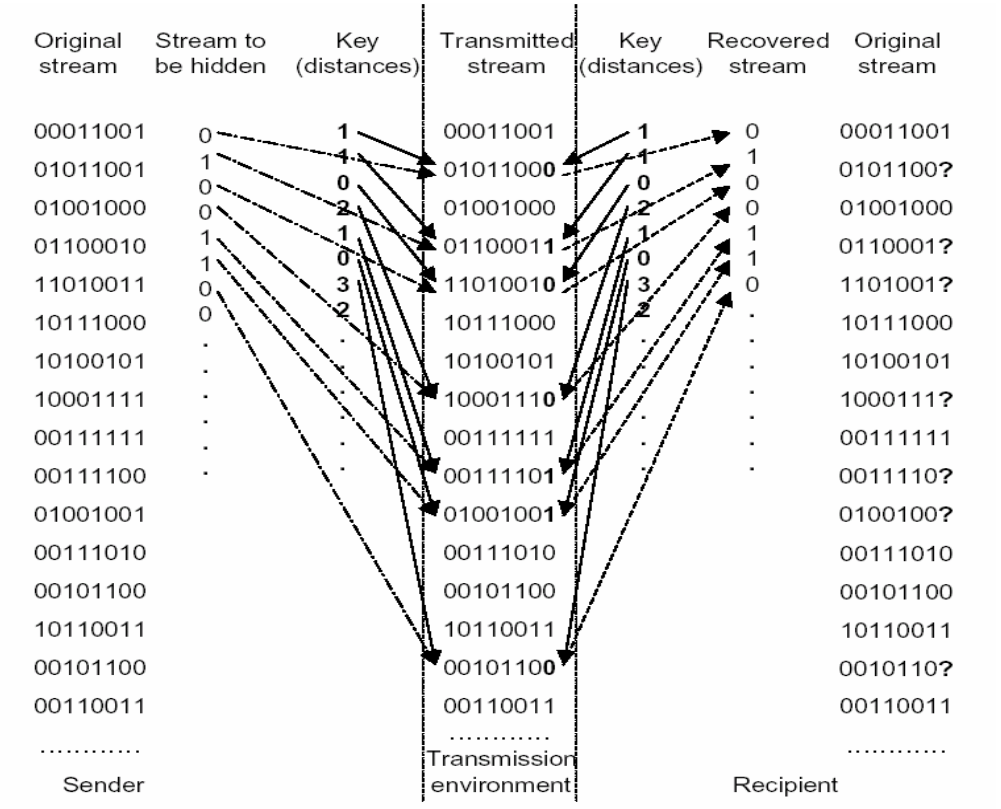

Figure 3: Example of LSB and scrambling [Popa98]

Furthermore, this watermarking algorithm provides an Error Correction Code (ECC) based on the Viterbi algorithm [GLO97]. If ECC is used, then the whole watermarking message size is doubled. The employment of ECCs provides the algorithm a mechanism to correct errors which can occur during transmitting or attacking the audio signal.
This watermarking algorithm is blind and does not require the original audio file to detect and retrieve the embedded watermark information. For detection it needs only the key (if used) and the knowledge of if error correction was used during the embedding process.

Parameters of the LSB watermarking algorithm (notation of the used implementation displayed here):

- m : watermarking message, which is embedded
- k : key to initialize the PRNG, which is used for the scrambling mode
- c : flag for ECC (binary value – either ON or OFF)

(a.3.2.2) Spread Spectrum

The Spread Spectrum scheme has been well studied in the watermarking literature ([BTH96], [CKLS96], [CKS01], [KM01], [Kim00], [Kim03], [LH00], [SHK02], [SZTB98]) and is the most popular watermarking scheme in use today. The algorithm spreads a pseudo-random sequence across the audio signal in frequency domain using a Fourier transformation.

The watermarking message is a binary message $m = \{0,1\}$ or an equivalent bipolar variable $m = \{-1,1\}$, which is modulated by a pseudo-random sequence $r(n_i)$. This sequence is generated by a secret key k. The value α specifies the embedding strength. The index i extends from 1 to N, where N is the length of the audio signal. The following equation (1) demonstrates how the watermarked audio is produced:

$$x(n_i) = s(n_i) + \alpha w(n_i) \qquad (1)$$

$x(n_i)$ - watermarked audio signal
$s(n_i)$ - original audio signal
α - scaling factor
$w(n_i)$ - modulated watermark

The scaling factor α controls the adjustment between robustness and inaudibility. The modulated watermark $w(n_i)$ is equal to $r(n_i)$ or $-r(n_i)$ depending on m = 0 or m = 1.

Spread Spectrum is a blind watermarking method, which means that it does not need the original audio to detect the embedded watermark information. For detection a linear correlation is used, exploiting the fact that the pseudo-random sequence $r(n_i)$ is known since it can be regenerated using the key *k* (See [Kim] for detailed explanations and equations). The watermark is detected by calculating the correlation between $x(n_i)$ and $r(n_i)$. The determination of a watermark being detected depends on the correlation value *c* and a predefined threshold *t*. The watermark message is detected if w = 1.

$$w = \begin{cases} 1 & \text{if } c > t \\ 0 & \text{if } c \leq t \end{cases} \qquad (2)$$

The threshold influences the detection rate of false positives (watermark detected, where no watermarking was embedded) and false negatives (no watermark detected, where a watermark was embedded).

Parameters for the Spread Spectrum algorithm (notation of the used implementation displayed here):

- m : watermarking message, which is embedded
- k : key to initialize the PRNG
- c : flag for ECC (binary value – either ON or OFF)
- l : lowest frequency bound
- h : high frequency bound
- a : embed strength

For detection the algorithm requires the key, the knowledge of if error correction was used during the embedding process and the lowest and highest frequency band, where the watermark was embedded.

(a.3.2.3) Publimark

Publimark [PubHt] is a command line tool which embeds a secret message into an audio file. It uses a pair of keys (a public and a private key) for the exchange of the used secret key for the symmetric steganographic scheme. The public key is shared so that anybody can send a secret message, while the private key must be kept secret so that only the owner (receiver) can detect and retrieve the hidden information.
The embedding process consists of two phases. First, the sender chooses a random key (denoted *seed*), which is required as secret key to embed the steganographic message. This key is encrypted with the shared public key of the receiver and embedded into the cover-signal. Second, the sender transmits the steganographic message in an audio file to the recipient using an efficient private key steganographic algorithm [GFD02] initialized with the choosen seed as secret key. For the hidden transmission of the public encrypted secret key the Scalar Costa scheme [EBTG03] is used in combination with trellis-coded quantization

[Gue05] to improve the undetectability in accordance to the stego-signal statistics and signal quality.
Parameters for the Publimark algorithm (notation of the used implementation displayed here):

- m : watermarking message, which is embedded
- $K_{Private}$: private (secret) key
- K_{Public} : public key

(a.3.2.4) AMSL Audio Water Wavelet (2A2W)
This scheme was designed and implemented at the University of Magdeburg and embeds the watermark signal in the wavelet based frequency domain of the cover audio using a digital watermarking technique called zerotree (ZT) [Sha93]. It is a non-blind method, which means that the algorithm requires additional information (specific file, where the wavelet coefficients used for embedding are stored) for watermark detection. A classification of which wavelet coefficients are significant when using zerotrees is performed in [IMYK99].

Also in [IMYK99] are descriptions of two methods for embedding the digital watermark. The first method uses the insignificant coefficients, into which the watermark information is embedded redundantly. For detecting the watermark the zerotree root is used after the wavelet decomposition (computation of median and differences of the audio signal and the haar wavelet). The second method uses the significant coefficients by thresholding and modifying these coefficients at the coarsest scale. Two thresholds T_1 and T_2 (where $T_1 < T_2$) and one of the sub-bands must be selected. The absolute coefficients used for embedding must lie between T_1 and T_2. The watermark information is then embedded by modifying the calculated coefficients. The embedded position and the threshold value are read from the position file generated by the embedding function in order to detect the watermark after the wavelet decomposition of the marked audio file. It is this requirement which makes 2A2W a non-blind watermarking algorithm.

Parameters for the 2A2W algorithm (notation of the used implementation displayed here):

- m : watermark message, which is embedded
- w : watermarking method (For this implementation, this can only be ZT)
- c : coding method (For this implementation, this can only be BIN (binary))

(a.3.2.5) Viper Audio Water Wavelet (VAWW)
This watermarking scheme was designed and implemented at Purdue University and doesn't use the original audio data for the detection of the embedded signal, and is therefore a blind procedure [DRA01].

For adding the watermark into the audio signal, a three level Discrete Wavelet Transform (DWT) and a Daubechies 8-tap filter is used [DRA01]. This removes the low pass sub-band leaving only the coefficients in the other sub-bands, which are determined by a predefined threshold T_1. The watermark is added only to the high pass band. Although the watermark is only added to the coefficients above T_1, an audio signal sized watermark is used (same length of audio signal and watermark message). This has the advantage that the watermark is fixed at a particular location in the DWT domain of the audio signal, which makes it independent from the order of the coefficients. The parameter *s* is a scalar factor where it is assumed that *s* scales the watermarking pattern over a larger region of the audio signal, thus having an effect on the transparency and robustness of the watermarked signal. For detection of the embedded signal a second threshold T_2 (where $T_2 \geq T_1$) is used. All high pass coefficients above T_2 are correlated with the original copy of the watermark. A limit of 50% of the original correlation

is used for deciding if the watermark is detectable or not. If the detector states that more than 50% correlates with the watermark, then the watermark is successfully detected.

Parameters for the VAWW algorithm (notation of the used implementation displayed here):

- k : key to initialize the PRNG
- t : threshold
- s : scale factor

(a.4) Test environment for the evaluation of audio watermarking algorithms

In this subsection, the test environment, chosen parameters for the watermarking algorithms and the methods to evaluate the watermarking algorithms using profile evaluation are introduced. The general formal description divides the profiles into three categories; basic (B), extended (E) or application (A). This designation identifies the type of evaluation and it is indicated by capital letters. The same notation as in [LDLD05] is used, where for example $P_{E\text{-}Capacity}$ indicates an embedding profile with a name *capacity* is used. In the following test, the basic profiles $P_{B\text{-}Robustness}$, $P_{B\text{-}Transparency}$, $P_{B\text{-}Capacity}$ and $P_{B\text{-}Complexity}$ are used for the evaluation.

(a.4.1) General test environment:

For the testing hardware from the AMSL (Advanced Multimedia and Security Laboratory of the Research Group for Multimedia and Security of the Otto-von-Guericke University of Magdeburg) was used. This included eight workstations (in the following referred to as nodes N_1 to N_8) on which the evaluation tests were run. Table A1 in Annex A, Section (a.3) lists all relevant information about those workstations (CPU and memory configurations; other parts of the hardware such as hard drives, Network Interface Cards (NICs), graphics hardware and operating system version do not effect the measured computation time, because the measuring method used for these tests counts only the used CPU time of the running process. Furthermore, a local copy of the audio files is used to reduce the network bandwidth during the evaluation tests). All the nodes described in this table were running the watermarking evaluation tests as a single task, with all other services disabled during this time.
All computations in the first round of evaluations required a computation time of about 342,810 seconds (3.97 days) for embedding, retrieval and attacking and 4,726,080 seconds (54.7 days) for the measurement of transparency evaluations ($P_{B\text{-}Transparency}$). This measure includes only CPU time spent on the actual evaluation process, with time for extraneous functions such as the operating system or any other process disregarded. The measurements were obtained using the Linux/Unix `time` command, using only the *user* value. Additional computation time was measured using data mining and analysis scripts. To satisfy the enormous need for computation power implied by these measurements, an additional workstation was acquired to improve the testing speed and capacity.

Figure 4 shows the general evaluation process. It can be divided into three parts: the Embedding Process, the Attacking Process and the Detection and Retrieval Process. All these processes require additional parameters, as described below.

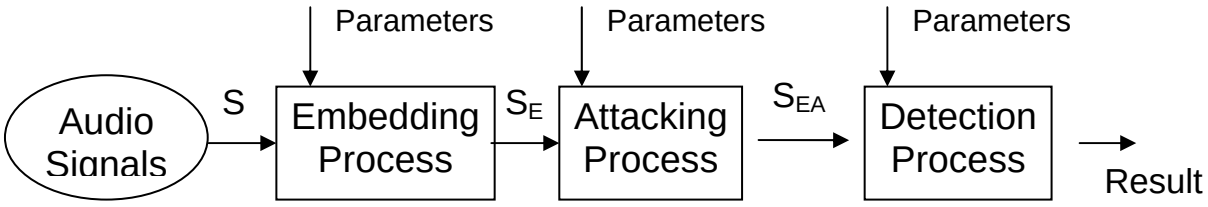

Figure 4: Evalution Process

(a.4.2) Parameters for the watermarking algorithms

The following tables show the disposition of the tasks on the nodes N_1 to N_8 to the watermarking algorithms, and the parameters used for the selected watermark algorithms (embedding process).

(a.4.2.1) Least Significant Bit

Table 1 shows the parameters used for the embedding process. All four combinations of ECC (on, off) and used key (yes, no) were evaluated. Four different nodes were used in the evaluation.

Node	Message (m)	Key (k)	ECC (c)
N_5	University of Magdeburg	22	yes
N_7	University of Magdeburg	Ø	yes
N_3	University of Magdeburg	22	no
N_6	University of Magdeburg	Ø	no

Table 1: Test parameters for the LSB watermarking algorithm

(a.4.2.2) Publimark

The Publimark algorithm was run on one computer, with a pre-generated key pair and the parameters shown in Table 2.

Node	Message (m)	P_{rivate}	P_{ublic}
N_4	University of Magdeburg	4,648 bit	1,024 bit

Table 2: Test parameters for the Publimark watermarking algorithm

(a.4.2.3) VAWW - Viper Audio Water Wavelet

The VAWW algorithm was run twice with different scalar values, shown in Table 3. One node was used for the evaluation of this algorithm.

Node	Key (k)	Threshold (t)	Scalar (s)
N_2	22	40	0.1
N_2	22	40	0.2

Table 3: Test parameters for the VAWW watermarking algorithm

(a.4.2.4) 2A2W - AMSL Audio Water Wavelet:

The 2A2W algorithms were run once on one node, and the parameters used are shown in Table 4.

Node	Message m	Key (k)	Encoding method (c)	Watermarking method (w)
N_8	University of Magdeburg	22	binary	ZeroTree

Table 4: Test parameters for the 2A2W watermarking algorithm

Spread Spectrum:

The Spread Spectrum algorithm was run four times in two different frequency bands, with and without ECC. The frequency bands used were 9-11 kHz and 17-19 kHz. The frequency band

of 9-11 kHz has been used in for testing in the past [LDSV05], prompting its use for this evaluation. The frequency range 17-19 kHz was selected due to it being close the audible frequency bound for humans. It was expected to demonstrate good transparency for these parameters.

An embedding strength of 5000 was used as the default value. For all embedding parameters ECC was enabled and disabled. Table 5 shows the parameters and the four nodes used in the evaluation.

Node	Message m	Key (k)	ECC (c)	Lower frequency bound (l)	Higher frequency bound (h)	embed strength (a)
N_3	University of Magdeburg	22	yes	9,000	11,000	5,000
N_1	University of Magdeburg	22	yes	17,000	19,000	5,000
N_5	University of Magdeburg	22	no	9,000	11,000	5,000
N_4	University of Magdeburg	22	no	17,000	19,000	5,000

Table 5: Test parameters for the LSB watermarking algorithm

The test performance of the four basic profiles ($P_{B\text{-}Robustness}$, $P_{B\text{-}Transparency}$, $P_{B\text{-}Capacity}$ and $P_{B\text{-}Complexity}$) [LDLD05] will now be detailed.

(a.4.3) Evaluation of Robustness - $P_{B\text{-}Robustness}$

The attacking process used for testing robustness was StirMark Benchmark for Audio (SMBA) [SMBA], which includes 40 attacks.[1] Each attack was used with its default attack parameters, which are detailed in [LDSV05]. After the attacking process, the watermarking algorithm used for embedding is used to attempt to detect and to retrieve the watermark information (detection process).

For our current application the robustness of watermarking algorithms is classified into three distinguishable classes: robust, fragile and non-determinable. A watermarking algorithm W_i is considered robust against an attack a_k if in less then 10% of all marked files the message or watermark embedded is not retrievable. W_i is considered fragile against the attack a_k if in more then 90% of all marked files the message or watermark embedded is not retrievable. The robustness of W_i against an attack a_k is considered non-determinable if the percentage of successful attacks lies between 10% and 90% of all audio files. The results for each of the algorithms W_i considered will be given in the following form: (number of a_k against which W_i is robust / number of a_k where the robustness of W_i is considered non-determinable / number of a_k against which W_i is fragile). An example for this form would be (3/27/10), which would indicate that the corresponding algorithm W_i is considered robust against three attacks, in 10 cases W_i is fragile and no definitive answer can be given for 27 attacks. Considering the test set size of 389 files, the thresholds of 10% and 90% are equivalent to 39 and 350 files respectively.

Another aspect of robustness is the context dependency of the evaluation process. As introduced in section (a.3) the audio test set consists of material from different classes (*music*, *speech*, *sounds* and *SQAM*) with a huge number of subclasses (*male speech*, *jazz*, *noise*, ...; see section (a.3)). The evaluation of the watermarking algorithms performed here allows for statements about the context dependency of attacks, i.e. it becomes visible which attack performs how well on which audio material.

(a.4.4) Evaluation of Transparency - $P_{B\text{-}Transparency}$

For the transparency evaluation three goals were identified.

- The first and most obvious goal is the determination of the transparency of the evaluated audio watermarking algorithms W_i. For this evaluation the embedding transparency of W_i is measured by computing the ODG (Objective Difference Grade)

[1] AddBrumm, AddDynNoise, AddFFTNoise, AddNoise, AddSinus, Amplify, BassBoost, Compressor, CopySample, CutSamples, DynamicPitchScale, DynamicTimeStretch, Echo, Exchange, ExtraStereo, FFT_HLPassQuick, FFT_Invert, FFT_RealReverse, FFT_Stat1, FlippSample, Invert, LSBZero, Noise_Max, Normalizer1, Normalizer2, Nothing, Pitchscale, RC_HighPass, RC_LowPass, Resampling, Smooth, Smooth2, Stat1, Stat2, TimeStretch, VoiceRemove, ZeroCross, ZeroLength1, ZeroLength2, ZeroRemove

between S_E and S (introduced in Figure 4) using the open source software tool EAQUAL (Evaluating of Audio QUALity, [EAQ02])[2]. The value for the transparency of W_i is computed as the arithmetic sum of the ODG values between the original and marked files from the test set.

- The second goal is the evaluation of the transparency when single attacks are used, and to determine their impact on the watermark message while considering the degradation of the quality of the audio material due to the modifications made by the attack. In this case, the ODG is computed between S_E and S_{EA}. An attack will be considered successful in this evaluation if the watermark is destroyed by the attack (i.e. it is not detectable/retrievable) and the modifications on the audio material are considered better than "perceptible, but not annoying" (i.e. the average of all ODG(S_{EA},S) for this attack is better than -1). The results presented here are for the attacks run with their default parameters [LS04]. A modification of the parameters or the usage of transparency enhancing methods (like psychoacoustic modelling) could improve these results.
- Another important aspect for this evaluation is that in certain cases the ODG value between S_{EA} and S can be better (meaning improved audio quality) than the ODG between S_E and S (ODG(S_{EA},S) > ODG(S_E,S)). This means that S_{EA} is considered by the evaluation software to be more like the S than S_E. The third goal of this evaluation is to consider this fact is for every watermarking algorithm. The attacks which lead to improved results are identified there.

Transparency enhancing methods, such as the modification of attacks using psychoacoustic modelling, were already discussed in theory in [LDS03]. [KRA05] and [DKL05] give the first evaluation results for a prototypical implementation of a psychoacoustic model. Further results will be presented in an upcoming publication presented at SPIE 2006 ([KDL06]). It is anticipated that significant improvements in the attack transparency can be achieved by modified attacks using psychoacoustic modelling.

(a.4.5) Evaluation of Capacity - $P_{B\text{-}Capacity}$

The capacity of the watermarking algorithms considered is measured in two different ways for this evaluation. First, pre-generated messages of increasing lengths are embedded iteratively into one pre-selected file (e.g. music___blues___BBKing-GuessWho.wav – chosen by random) until the algorithm is no longer capable of embedding and retrieving the message correctly. The results will be presented relative to the file-size of the test file chosen (absolute results can be found in Annex A (a.3), Table A2). The second way of determining the capacity of an algorithm is the usage of a pre-defined message of fixed length (e.g. "UniversityOfMagdeburg", length: 21 characters) to qualify the payload. For the algorithms where it is possible to embed messages more than once, the number of successful embeddings into the file (such as music___blues___BBKing-GuessWho.wav, the example above) is counted. From this number, a payload in bits per second is derived.
The difference between the first and the second way of determining the capacity is the number of synchronisation blocks used. In the first case, one large message (therefore one synchronisation block) is used to determine the maximum theoretical capacity of a file of given size. In the second case, the number of messages of a fixed length which can be embedded (equalling the number of synchronisation blocks) is measured.

[2] The ODG lies on a scale ranging from 0 (imperceptible) to -4 (very annoying).

(a.4.6) Evaluation of Complexity - $P_{B\text{-}Complexity}$

For the determination of the complexity of W_i, the embedding and retrieval time[3] is measured. The results are given with their minimum, maximum and average values. The tests were run on eight different workstations (nodes N_1 to N_8 as described in Section (a.3) of this report and in Annex A, Table A1). To make the results comparable, a scaling matrix based on the performance of the nodes running the actual watermarking algorithms was used to normalize the values.

In Annex A, Section (a.3), Table A3 and Table A4 show the un-normalized computation times, which are the times needed by the CPU to perform the embedding, attacking or detecting process. In this work the complexity of a process is defined as the measured time this process spends on the CPU. If multiple nodes are used, the results have to be normalized to provide comparability. Table A5 lists the scaling factors for each of the algorithms, and Table A6 shows the overall scaling factors used to compare the complexities.

(a.5) Test results

In this subsection, the test results are introduced for the evaluation of audio watermarking algorithms. The evaluation of $P_{B\text{-}Robustness}$, $P_{B\text{-}Transparency}$, $P_{B\text{-}Capacity}$ and $P_{B\text{-}Complexity}$ [LDL05] is introduced first with global test results, followed by test results for each individual watermarking algorithm.

(a.5.1) Global test results

Table 5 summarizes the overall test results regarding the algorithms and the evaluated aspects of $P_{B\text{-}Robustness}$, $P_{B\text{-}Transparency}$, $P_{B\text{-}Capacity}$ and $P_{B\text{-}Complexity}$. In this table the basic results for all watermarking algorithms are presented to allow for a comparison of the algorithms. More detailed descriptions of the results and additional measures can be found in the following sections, where the evaluation results for each algorithm are presented separately.

As mentioned in Section (a.3), one degenerated file influenced the test results. This file is the cause of the minimum embedding and retrieval times of 0 seconds for some algorithms in Table 12. This degenerated file also influences the robustness and transparency evaluations, leading to 41 (out of 15,959) cases were no watermark can be embedded (and retrieved) in the robustness evaluation. When considering the transparency, no ODG value for this file can be computed. The influence of this single degenerated file is small enough (0.257% of the test set) to be neglected in the following considerations.

[3] Only the computation time of the CPU for the corresponding algorithm is considered; this figure contains no computation time for the operating system or other processes.

Algorithms	Parameters	$P_{B\text{-}Complexity}$						$P_{B\text{-}Robustness}$	$P_{B\text{-}Transparency}$	$P_{B\text{-}Capacity}$	
		embed			retrieve					average	
		min	max	avg	min	max	avg			payload (Byte/s)	embedding capacity
Least Significant Bit	key=22 ECC ON	0	0.131	0.119	0	26.224	10.728	5/7/28	0.00100	1,052.417213	< 1%
	key={} ECC ON	0	0.463	0.171	0	48.992	20.556	7/9/24	-0.00296	5,261.342306	< 1%
	key=22 ECC OFF	0	0.273	0.17	0	0.6646	0.3408	5/7/28	0.00087	2,104.834425	1,3%
	key={} ECC OFF	0	0.445	0.171	0	3.6144	1.5417	3/7/30	-0.00201	10,523.42837	< 1%
Publimark		7.255	9.609	8.38	0.18	0.2181	0.2055	3/6/31	0.01486	n.a.	2,1%
Spread Spectrum	ECC ON High	0	4.2	1.698	0	2.1462	0.9264	0/17/23	-0.68059	0.743757747	< 1%
	ECC ON Middle	0	4.89	2.045	0	2.0278	1.0224	0/19/21	-2.24794	0	< 1%
	ECC OFF High	0	3.042	1.263	0	1.6302	0.6888	0/16/24	-0.81198	2.97503099	< 1%
	ECC OFF Middle	0	3.588	1.43	0	1.9072	0.8344	0/20/20	-2.38005	0	< 1%
VAWW	s=0.1	0.123	8.915	2.152	0.12	4.7801	1.2296	20/8/12	-1.89265	n.a.	n.a.
	s=0.2	0.108	8.976	2.152	0.12	4.4266	1.2296	19/8/13	-2.73116	n.a.	n.a.
2A2W		0.031	0.979	0.217	0	0.3329	0.074	7/7/26	-2.80000	n.a.	< 1%

Table 5: Overall test results

(a.5.2) Least Significant Bit – Test Results

In this subsection test results for the LSB watermarking algorithm are presented. The full results for all of the tests run on the LSB watermarking algorithm can be found in Annex A, Section (a.3), Figures A3-A34 and Tables A7-A18. As described in Section (a.3) of this report, the LSB algorithm introduced was run with four different parameter sets (key on and off, ECC on and off).

a) Robustness

The robustness of the LSB watermarking algorithm against the attack suite ranges for the four different parameter sets, from (3/7/30) in the worst and (7/9/24) in the best case. The parameters used and the corresponding results can be found in Table 12. In the case without key and with ECC enabled demonstrates the best result in terms of robustness, while the worst result is in the instance where no key is used and with ECC disabled.

Considering the robustness against selected attacks, it can be stated that the LSB watermarking algorithm can be considered robust (in every parameterization) against CopySample, Invert and Nothing. In three out of four cases it is also considered robust against Compressor and FlippSample. For more details see Annex A, Section (a.3), Tables A7, A10, A13, A16.

Table 6 shows that no clear context dependency of the evaluation process can be identified.

music										sounds				Speech				SQAM		
blues	classical	country	hiphop	jazz	metal	pop	reggae	synthetic	techno	computergen	natural	noise	silence	computergen	female	male	sports	instrumental	voice	Parameters
16.8	15.7	18.0	15.9	15.6	18.0	16.4	17.8	20.4	17.9	15.8	12.8	20.2	20.0	19.4	13.8	12.8	15.2	21.4	21.9	ECC on, key
17.2										17.2				15.3				21.7		
29.1	28.4	30.0	26.6	27.5	30.3	27.4	30.9	34.0	30.8	25.8	25.0	28.4	23.8	35.4	22.0	22.8	26.4	40.0	37.8	ECC on, no key
29.5										25.7				26.6				38.9		
16.9	15.9	17.4	14.9	15.4	16.9	17.1	17.8	20.5	17.5	13.3	15.0	19.8	21.3	19.3	13.1	13.3	15.5	21.4	20.8	ECC off, key
17.0										17.3				15.3				21.1		
16.4	17.1	16.9	16.4	15.6	16.4	17.9	17.0	16.9	17.0	12.1	17.5	17.0	18.8	18.4	16.9	16.9	16.8	17.5	18.6	ECC off, no key
16.7										16.3				17.2				18.1		

Table 6: Context dependency - Successfully retrieved watermarks in percent

b) Transparency

The transparency of the LSB algorithm can be considered to be very good. Even in the worst of the four cases the ODG value of -0.00296 is deemed “imperceptible.”

The following 8 attacks (used with their default parameters) are considered to be successful against the LSB algorithm: AddBrumm, FFT_HLPassQuick, FFT_Invert, LSBZero, RC_HighPass, RC_LowPass, Stat1 and Stat2. All four different parameter sets show the

same results in this matter. From a global point of view these attacks are different in its nature. To provide an answer as to why these attacks are successful would require additional research, such as a detailed analysis of the internals of the LSB approach itself and a study on the impact of each attack to the LSB watermarking pattern.
Only in the case of the AddBrumm attack are the results for ODG(S_{EA},S) better than ODG(S_E,S) on a substantial number of files watermarked using the LSB algorithm:

- 135 for LSB with ECC and with key
- 141 for LSB with ECC and without key
- 133 for LSB without ECC and with key
- 137 for LSB without ECC and without key

For details on the performance of these attacks on files marked by the LSB algorithm, the reader is referred to Annex A, Section (a.3).

c) Capacity

With an average embedding capacity of 1.3% (62,274 out of 4,980,654 bytes) in the best case, the LSB watermarking algorithm has the second highest embedding capacity of all algorithms evaluated here. The capacity measured in the two cases where no key is given were obscured by problems in the retrieval of the watermark. Here, degenerated messages containing control characters disrupted the reading process and caused low capacity measures. The payload measured for the LSB watermarking algorithm ranges from an average of 1,052.4 bytes per second to 10,523.4 bytes per second, depending on the parameters. From the tests it is obvious that the figures for the algorithm without ECC are twice as high as the figures with ECC. It is also noticeable that the key has a strong influence on the payload (see Section a.3.2.1). The reader is referred to Annex A, Section (a.3), Table A2 for more details on the capacity evaluation for this algorithm.

d) Complexity

With average times between 0.119 and 0.171 seconds for embedding, the LSB watermarking algorithm is the fastest algorithm if only the embedding step is considered. The performance is distinctly higher in the cases where a key is employed than in the cases where one is not. The use of ECC seems to have little or no impact on the complexity of an embedding process.
In the case of message retrieval operations, the LSB algorithm requires distinctly more computation time as compared to embedding (between 0.3 and 21 seconds). Here, a clear difference in the computation times between the algorithm using ECC and without ECC can be seen. The usage of ECC (and without a key) improves the robustness against the single attacks FlippSample and CutSample. Both attacks are modification attacks that work in time domain. Therefore, usage of ECC can be considered useful if such attacks are expected.

(a.5.3) Publimark

The test results for the Publimark watermarking algorithm are presented here. The tables listing the actual test results for all tests run on the algorithm can be found in Annex A, Section (a.3), Figures A35-A42 and Tables A19-A21.

a) Robustness

The robustness of Publimark against the attack suite used can be given as (3/6/31). This is interpreted as the algorithm being considered robust against three attacks (Compressor, CopySample and Nothing), fragile against 31 attacks and in six cases no definitive answer can be given.

Table 7 shows that no clear context dependency of the evaluation process can be identified.

music										sounds				speech				SQAM	
blues	classical	country	hiphop	jazz	metal	pop	reggae	synthetic	techno	computergen	natural	noise	silence	computergen	female	male	sports	instrumental	voice
11.6	10.4	11.9	10.0	10.1	11.9	11.5	12.5	14.4	10.5	7.5	8.4	13.4	15.0	13.5	8.3	7.5	9.8	15.0	14.7
11.5										11.1				9.8				14.9	

Table 7: Context dependency - Successfully retrieved watermarks in percent

b) Transparency

The transparency of this algorithm can be considered to be very good. The overall ODG value of 0.01486 is considered imperceptible.
The following 10 attacks (used with their default parameters) are considered to be successful against Publimark: AddBrumm, BassBoost, FFT_HLPassQuick, FFT_Invert, Invert, LSBZero, RC_HighPass, RC_LowPass, Stat1 and Stat2.
In the case of Publimark no significant improvements of the ODG values after an attack can be found.

For details on the performance of attacks on files marked by Publimark, the reader is referred to Annex A, Section (a.3).

c) Capacity

With an average embedding capacity of 2.1% (103,700 out of 4,980,654 bytes) Publimark has the highest embedding capacity of all algorithms evaluated. However, no payload value could be determined for this algorithm, as it does not permit the embedding of more than one message. The reader is referred to Annex A, Section (a.3), Table A2 for complete details on the capacity evaluation for Publimark.

d) Complexity

Publimark requires the highest computation time for embedding of all the algorithms evaluated in this project. With an average of 8.38 seconds for each file, the time required is more than four times the computation time required by the Spread Spectrum algorithm described in the next section. However, when Publimark is used to retrieve a message it requires an average of only 0.2 seconds.

(a.5.4) Spread Spectrum

The test results for the Spread Spectrum watermarking algorithm are presented here. The tables listing the test results for all tests run on the Spread Spectrum watermarking algorithm can be found in Annex A, Section (a.3), Figures A43-A74 and Tables A22-A33.
As described in Section (a.3) of this report, the Spread Spectrum watermarking algorithm was run with four different parameter sets. The test results introduced here are not comparable with other spread spectrum watermarking algorithms.

a) Robustness

Problems with the watermark retrieval function of this algorithm lead to very poor results in the robustness evaluation[4]. Independent from the parameters used, the algorithm can not be considered robust against any attack. From the 389 original files in the test set the watermark could only be retrieved (after a successful embedding) in 107 to 249 cases, depending on the parameters used (See Table 5). Surprisingly, the number of successfully detected watermarks after VoiceRemove, RC-Highpass and ExtraStereo attacks was significantly higher than the number of successfully detected watermarks on marked but non-attacked files. The signal modification performed by the attacks improved the strength of the embedded watermark, leading to a better detection rate. The reader is referred to Annex A, Section (a.3) for more details.

Table 8 shows content dependencies of the evaluation process. Three different classes of dependencies can be identified here, which should be analyzed in detail in following research work:

- Sub-Category dependence: For complete sub-categories (in this case, music/hiphop and sounds/computergen) the attack suite used performs very well compared to the overall set. In these cases the watermark is still retrievable after the attacks in less then two percent of all cases. These instances are marked in light grey in Table 8.
- Frequency band dependence: In ten cases the attacks on the algorithm perform significantly better in the frequency range of 9 to 11 kHz (Middle) than in the range of 17 to 19 kHz (High). These instances are marked in medium grey in Table 8.
- ECC dependence: In two cases (sub-categories sounds/silence and speech/computergen) the attacks on the algorithm performed significantly better if ECC is enabled. These instances are marked in dark grey in Table 8.

music										sounds				speech				SQAM		
blues	classical	country	hiphop	jazz	metal	pop	reggae	synthetic	techno	computergen	natural	noise	silence	computergen	female	male	sports	instrumental	voice	Parameters
32.4	31.6	23.6	1.1	32.4	2.8	9.3	12.0	27.5	3.3	0.0	24.4	9.5	23.8	4.0	10.3	24.1	21.4	16.4	11.4	ECC on, High
17.6										14.4				14.9				13.9		
12.0	29.6	8.3	0.0	27.5	0.0	0.1	2.8	9.8	0.4	0.0	8.4	9.1	26.3	0.6	0.9	7.9	13.4	13.6	12.5	ECC on, Middle
9.0										10.9				5.7				13.0		
31.4	33.4	23.5	0.8	30.0	6.3	8.6	12.9	31.4	4.6	0.0	25.6	10.2	47.5	24.1	19.1	33.0	29.3	25.0	28.1	ECC off, High
18.3										20.8				26.4				26.5		
13.4	34.1	5.0	0.0	27.0	1.3	0.1	2.8	13.0	4.3	0.0	13.1	11.6	56.3	14.9	2.5	5.1	14.1	20.7	11.4	ECC off, Middle
10.1										20.2				9.1				16.1		

Table 8: Context dependency - Successfully retrieved watermarks in percent

[4] The algorithm had problems to detect sync marks even in non-attacked files

b) Transparency

The four transparency evaluation results pertaining to this algorithm can be divided into two groups:

- Results for the algorithm run in a high frequency band (17 to 19 kHz; ODGs -0.68059 and -0.81196)
- Results for the algorithm run in a medium frequency band (9 to 11 kHz; ODGs -2.24794 and -2.38005).

The results in the high frequency band are noticeably better then the ones in the lower frequency band due to the psycho-acoustical properties of the ODG measure[5]. Furthermore, it is apparent from the data that the results without ECC are more transparent than the results with ECC, a result of the larger amount of data in the later case.

The following 3 attacks (used with their default parameters) are considered to be successful against Spread Spectrum if the frequency range of 17 to 19 kHz (High) is used: FFT_HLPassQuick, Stat1 and Stat2. If the frequency range of 9 to 11 kHz (Middle) is used no attack can be considered successful, due to the poor ODG results in these cases.

In the case of the Spread Spectrum algorithm, global improvements of the ODG values after an attack can be found in AddBrumm, RC_LowPass, Stat1 and Stat2 attacks. When the parameters ECC=on and frequency range=high are used an additional improvement in FFT_HLPassQuick can be found. Likewise, when the parameters ECC=off and frequency range=high are used an additional improvement in FFT_HLPassQuick, BassBoost and Invert can be found. When the parameters ECC=on and frequency range=middle are used an additional improvement in Amplify, FFT_Invert and LSBZero can be found. The most improvements occur if the parameters ECC=off and frequency range=middle are used: AddSinus, Amplify, BassBoost, FFT_HLPassQuick, FFT_Invert and LSBZero.

For details of the performance of attacks on files marked by Spread Spectrum, the reader is referred to Annex A, Section (a.3).

c) Capacity

With an embedding capacity of less then 1%, Spread Spectrum has a very low embedding capacity. Problems encountered while using this algorithm (in particular, problems detecting the synchronization pattern correctly and reading degenerated messages) made the capacity evaluation difficult, leading to poor results. An attempt to determine the payload capacity resulted in a range from 0 bytes per second in the worst case (i.e., it was not possible to embed and retrieve a message successfully) to 2.975 bytes per second in the best case. The reader is referred to Annex A, Section (a.3), Table A2 for more information.

d) Complexity

With averages between 1.263 and 2.045 seconds for embedding and 0.7 to 1 seconds for watermark retrieval, the computation time for the Spread Spectrum lies in the middle of the test field. Two facts worth noting are that the time for embedding is about the double time for retrieval. Without ECC the algorithm the (embedding and retrieving) works slightly faster.

[5] The human auditory system, which is modeled in the ODG computation, is far more sensitive to changes in the center of the audible field than on the borders.

(a.5.5) Viper Audio Water Wavelet - VAWW

In this section the test results for the VAWW watermarking algorithm are presented. The tables listing the actual test results for all tests run on the algorithm can be found in Annex A, Section (a.3), Figures A75-A90 and Tables A34-A39.

a) Robustness

According to the form described above, the robustness of the VAWW[6] algorithm against the attack suite is (20/8/12) in the case that the parameter *s* = 0.1 and (19/8/13) in the case that the parameter *s* = 0.2.

Table 9 illustrates two different facts:

- A single context dependency concerning the sub-class sounds/computergen can be identified (Table 9, depicted in gray). The algorithm appears to be more fragile against the attack set for this sub-class.
- The parameter *s* has no significant influence on the robustness of the algorithm.

Music										sounds				Speech				SQAM		
blues	classical	Country	hiphop	jazz	Metal	pop	reggae	Synthetic	techno	Computergen	natural	noise	silence	computergen	female	Male	sports	instrumental	voice	Parameters
63.8	61.8	66.8	64.0	63.1	65.0	65.5	63.5	70.6	62.0	13.3	65.9	56.8	27.5	54.3	54.8	58.9	57.0	23.6	55.8	
64.6										40.9				56.2				39.7		s=0.1
62.6	60.8	64.6	63.4	62.1	64.6	64.9	62.9	66.9	62.0	12.9	65.6	56.4	27.5	54.0	54.3	58.5	56.6	23.2	55.3	
63.5										40.6				55.8				39.2		s=0.2

Table 9: Context dependency - Successfully retrieved watermarks in percent

b) Transparency

The average transparency of this algorithm can be stated as ODG = -1.89265 in the case of *s* = 0.1 and ODG = -2.73116 in the case of *s* = 0.2. These values are considered to be in the range between “perceptible, but not annoying” (-1.0) and “annoying” (-3.0).

For this algorithm no attack can be considered successful (due to the bad ODG results in all cases).

For the VAWW algorithm, global improvements of the ODG values after an attack can be found in AddBrumm, Amplify, RC_LowPass, Stat1 and Stat2. A higher number of improved cases can be found if the parameter *s* is set to 0.2. The average ODG values decreased for AddBrumm from -1.87 (*s* = 0.1) to -2.68 (*s* = 0.2), for Amplify from -1.92 to -2.51, for RC_LowPass from -1.77 to -2.58, for Stat1 from -1.03 to -1.51 and for Stat2 from -1.31 to -1.94. Based on the assumption stated in Section a.2.5, the scaling factor *s* has an influence on the transparency of the watermarked audio signal, and this is illustrated here. If *s* is increased, then the transparency is worse.

For details of the performance of these attacks on files marked by the Spread Spectrum algorithm, the reader is referred to Annex A, Section (a.3).

[6] A watermark is considerd detected by VAWW if the absolute value of the returned correlation value is larger than 50%.

c) Capacity

The VAWW watermarking algorithm does not allow for the embedding of user-defined messages. Instead it computes an own watermark based on the characteristics of the audio material. Therefore, no capacity could be measured using the methods laid out for this evaluation. Additionally, no payload value could be determined for this algorithm because it does not permit the embedding of messages.

d) Complexity

With an average computation time of 2.152 seconds for each file, the VAWW needs the second longest embedding time of all algorithms. Retrieval works faster then embedding, clocking in at about half that time. Furthermore, the parameter *s* seems to have no influence on either the embedding or the retrieval time.

(a.5.6) AMSL Audio Water Wavelet (2A2W)

In this section the test results for the 2A2W watermarking algorithm are presented. The tables listing the actual test results for all tests run on the algorithm can be found in Annex A, Section (a.3), Figures A91-A98 and Tables A40-A42.

a) Robustness

According to the form described above, the robustness of the 2A2W algorithm against the attack suite used can be stated as (7/7/26), meaning this algorithm is considered robust against seven attacks (AddBrumm, AddSinus, Amplify, ExtraStereo, LSBZero, Nothing and VoiceRemove), fragile against 26 attacks, and in seven cases no definitive answer can be given.

Table 10 shows the content dependencies uncovered in the evaluation process. Two different classes of dependencies are identified here:

- Category dependence: For a complete category (in this case speech) the attack suite used performs significantly worse than for all other categories. This is can be seen in Table 10 marked in dark grey.
- Sub-Category dependence: For a complete sub-category (in this case sounds/silence) the attack suite used performs very well, with every watermark being destroyed. This is denoted in Table 10 marked in light grey.

music										sounds				speech				SQAM	
blues	Classical	country	hiphop	Jazz	metal	pop	reggae	synthetic	techno	Computergen	natural	noise	silence	computergen	female	male	sports	instrumental	voice
22.3	23.0	22.4	18.6	23.5	16.6	20.0	21.6	21.3	18.8	25.6	26.3	20.7	0.0	33.3	35.3	35.7	35.0	24.3	27.5
20.8										18.1				34.8				25.9	

Table 10: Context dependency - Successfully retrieved watermarks in percent

b) Transparency

With a general ODG value of about -2.8, the average output of the algorithm is almost “annoying”. For this algorithm no attack can be considered successful due to the bad ODG results in all cases.

In the case of the AddBrumm (320), RC_LowPass (260), Stat1 (240) and Stat2 (205) attacks on files watermarked with 2A2W the results for ODG(S_{EA},S) were better than ODG(S_E,S) on a significant number of files.

For details of the performance of these attacks on files marked by the 2A2W algorithm, the reader is referred to Annex A, Section (a.3).

c) Capacity

The capacity of the programme was limited by the developers to 101 bytes. Therefore, no real capacity measurement on this algorithm could be performed. No payload value could be determined for this algorithm because it does not permit the embedding of more than one message.

d) Complexity

The 2A2W algorithm requires an average computation time of 0.22 seconds for each file when embedding. If 2A2W is used to retrieve a message it requires an average of only 0.074 seconds.

Summary and Conclusion of the Test Results

Below is a summary of the test results by algorithm, followed by a comparison of the algorithms in terms of their complexity, robustness, transparency and capacity.

- The evaluation of the LSB watermarking algorithm shows that the complexity increases by using ECCs in the retrieval process. If a key is employed, then the complexity for the embedding process increases from 0.119 to 0.171 by using ECC. The usage of a key and/or ECC does not have an effect on the robustness or transparency, which are independent of these parameter settings. When using ECC, the capacity is reduced by half. If a key is used in addition, then the capacity is about a fifth (due to the randomly chosen jumping of the marking positions, which has a maximum of 10).
- The Publimark algorithm was evaluated with one key pair, and no other parameters are possible.
- The different embedding parameters used here for the Spread Spectrum algorithm demonstrate that the complexity (time to embed) increases if the watermark information is embedded in the middle frequency range. This watermarking algorithm is not very robust and we expect a problem in the design or implementation of it. Depending on the embedding frequency range, the transparency is improved if the high frequency range is used.
- The different embedding parameters for the VAWW algorithm show that the transparency depends on the scalar parameter value *s*. If this value is increased, then the transparency is decreased. The other basic profiles do not show such a direct effect by the different embedding parameters.
- The 2A2W algorithm was tested with one parameter setting, due to the fact that there are not parameters available which can have an effect on the results of the basic profiles.

If the evaluated watermarking algorithms are compared with each other, it was shown that the complexity of the embedding depends on the working domain of the embedding algorithms while the complexity of the retrieval process depends on the use of ECCs. The robustness of the evaluated watermarking algorithms shows that the VAWW algorithm provides the highest robustness against single attacks. The transparency of the embedding process of the evaluated watermarking algorithms shows that the watermarking algorithms working in wavelet domain have the worst transparency, while the LSB and Publimark algorithms have the best

transparency. Since the capacity is not able to be measured for each of the watermarking algorithms used, a comparison is difficult. Out of the evaluated watermarking algorithms on which capacity calculations were able to be made, it was shown that the use ECC decreases the embedding capacity (as expected). These results are important in performing a tradeoff analysis when designing watermarking systems.

(b.1) Design, test objectives and hypotheses for the VoIP application

Based on the knowledge and experiences from existing image steganography and steganalysis techniques, the overall objective of Task (b) is to design and implement audio steganography in ad-hoc, end-to-end media communications (streaming application using packet-based communication). To meet this end the example of Voice over Internet Protocol (VoIP) was selected, and an audio steganalysis framework was built with a special focus on cover-stego attacks. There are two types of steganalysis for VoIP designed, an active and a passive communication. The active VoIP scenario means that the communication partners itself transmit the secret message. In contrast, for the passive VoIP scenario, the sender and receiver of the secret message use an existing VoIP stream (by inject the secret massage) from other communication partners to hide there data. A possible attacker cannot distinguish, if the communication contains a secret message or not. The overall design is performed for both VoIP scenarios and the focus of the implementation is set only on the active VoIP scenario. Jori's Voice over IP library by Jori Liesenborgs (JVOIPLIB) is therefore used, as it provides primitives for a basic VoIP communication. The general design of the VoIP steganography algorithm is based on known LSB hiding techniques (used for example in StegHide (http://steghide.sourceforge.net/ (2004), see details in [DHH05]). For the overall task, the properties of the introduced steganographic method within VoIP regarding subjective transparency evaluation, objective transparency evaluation, steganalysis, as well as reliability, have been analyzed. A VoIP communication tool based on JVOIPLIB with integrated steganographic embedding and retrieval function has been implemented to enable this evaluation. For an analysis of the performance of the communication tool, a steganalysis tool capable of detecting active VoIP communication, storing transmitted content in a database and performing window-based (1024 samples) analysis using 13 statistical attacks has also been developed ([DiHe04]).

For the VoIP scenario, three testing goals were defined, which were first described and published in [VDHK06]. First, imperceptibility was tested by applying both subjective and objective perception methods; these results were published in [KDL06]. Second, a self-developed steganalysis framework was employed to determine out how hard it is to detect the embedded steganographic message, based upon the steganographic payload used. Furthermore, the error rate with respect to the transmitted message was analyzed, and the reliability of the implemented software was investigated by using it non-stop over many hours. For this final test, the long time profile $P_{E\text{-}Long\_Time}$ [LDLD05] was applied in the test environment.

The test set for the VoIP application will now be introduced. Pre-recorded audio material is used in the testing instead of a "real" telephony session, due to the long time tests. These tests were to run for 240 hours without interruption, and for this duration the generation of a representative signal by human speakers could not be guaranteed. The pre-recorded audio material used was taken from the MIT audio database [GLF+93], which contains more than 2,300 English speech samples, and from the AMSL audio classification test set which consists of 389 audio files of speech, music and sound/noise signals (See Figure A2 in Annex A, Section (a.1) for more details on the AMSL set). This resulted in a set of 2,722 sound files,[7] which were played in random order. The audio player XMMS[8] was employed for testing on a SuSE-Linux platform, kernels 2.6.5-7.95-smp and 2.6.8-24.

[7] For the transmission via the VoIP channel all files used had to be resampled to 8 Bit signals with a sampling rate of 8000 Hz.

[8] X Multimedia System, http://www.xmms.org, 2005

Prior to performing the tests three test hypotheses were defined, which represent the expected test outcomes. Test results in Section (b.3) show the correctness of these hypotheses.

1. Hypothesis: For the minimum payload (package usage = 1%, which results in 2 bit per packet, or 20 ms Audio), transparency will be the highest. For that minimum payload it is very likely that human perception is not able to distinguish between cover and marked audio material. This will be measured through subjective and objective testing using the Objective Difference Grade (ODG), which is expected to show very good numeric results (near 0), based on a test set of 14 files.
2. Hypothesis: Computer-aided steganalysis, represented by the 13 statistical attacks of the steganalysis framework, will not be able to reliably detect steganographic messages in VoIP packets, based on a test set of 14 files.
3. Hypothesis: This VoIP implementation will satisfy the long time profile ($P_{E\text{-}Long\_Time}$), which will be tested with the complete test set of 2,722 sound files. Furthermore, the error rate η with respect to correctness of the transmitted steganographic message will be measured to demonstrate the reliability of the software. (Note that overall reliability depends on additional aspects, such as network traffic, routing protocols, performance of clients and routers). Overall, for the test environment η is expected to be very low ($\eta < 1\%$).

(b.2) Test environment and setup for the VoIP application

To enable testing of the active VoIP steganography scenario, a test environment consisting of three workstations connected via a 100 Mbit Ethernet was set up (Figure 5). A "star" network topology was chosen for the configuration, with a 100 Mbit hub to permit a sniffing of the VoIP connection. Two workstations act as the sender (A) and receiver (B), while the third workstation (C) is used to generate the audio signal used in the VoIP communication and, concurrently, to act as the attacker (IDS). For playback and recording purposes the anechoic chamber of Advanced Multimedia and Security Laboratory[9] (AMSL) and corresponding audio equipment was used (Figure 6). This includes a Shure SM-58 microphone and a Behringer Ultragain Pro Mic 2200 pre-amp.

[9] Research Group Multimedia and Security, Department of Computer Science, Institute of Technical and Business Information Systems, Otto-von-Guericke-University Magdeburg, Germany.

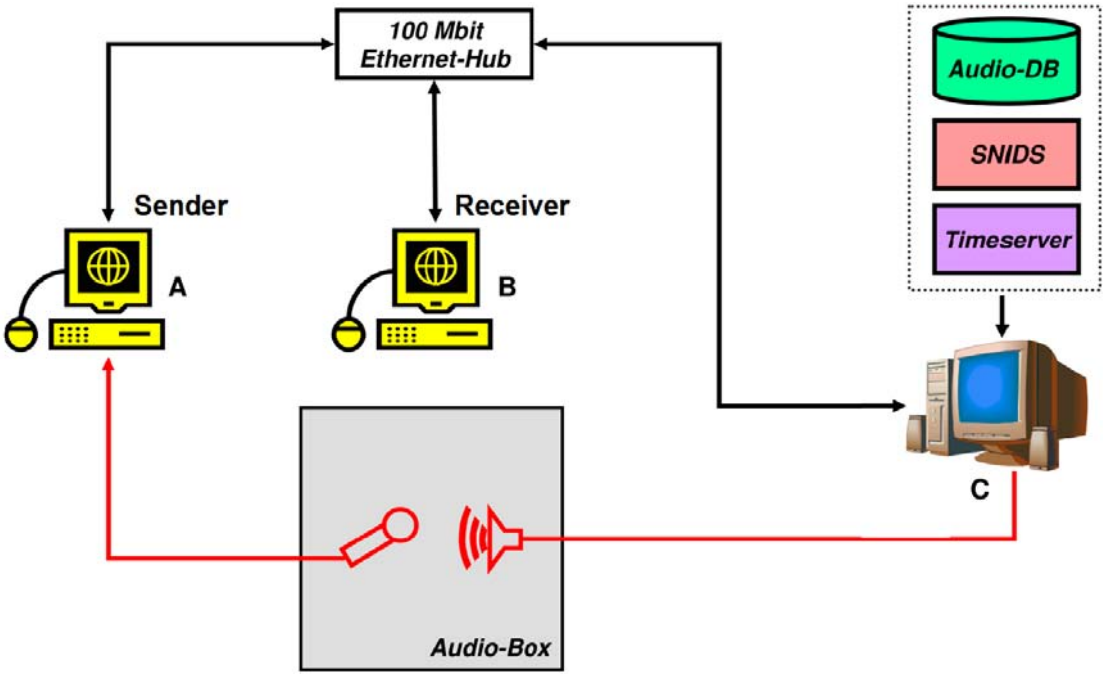

Figure 5: The test environment.

Figure 6: The anechoic chamber with its connection and control panel.

The analogue sound signal is played back by loudspeakers in the anechoic chamber and recorded with a microphone. This is done in order to simulate "real-life" speech and sound generation, with phenomena like analogue amplification, hissing and noise introduced during the recording by the microphone and the analogue sound processing and transmission. The signal provided is recorded by workstation (A) of the VoIP scenario and used as the input to the VoIP communication.

For testing, the VoIP software is set to unidirectional mode, where workstation (A) is always the sender and workstation (B) is always the receiver. This is achieved by running the software on receiver (B) in test mode (parameters: `-r -t`). The sender (A) also runs in test mode (parameter: `-t`), which is defined so that the embedding of the steganographic message will start automatically after three seconds run-time. After the message to be sent is transmitted successfully, the VoIP programme runs another three seconds before it also terminates. During this test session no user interaction (as would be required in "normal"

mode) is requested. Shell scripting is used to concatenate the transmission and processing of multiple messages.

Both sender (A) and receiver (B) generate log-files with time-stamps[10], the names of the transmitted files, a hash-value used for error detection and (on the sender side) the payload-factor used and the file size. This logging and the error detection mechanism are required to determine the influence of the message to be embedded on the transmission error.

For the subjective tests (hearing tests) a message of 100 Kbytes size was embedded using the off-line embedding function on 14 randomly selected files from the audio test-set. The payload-factor for those files was set to 1%, 25%, 50%, 75% and 100% respectively, resulting in 70 marked files, 14 original files and 15 minutes audio material for the subjective tests. In the hearing tests the listeners are asked to decide whether a signal presented is a marked or unmarked audio file. All listeners were students between 18 and 32 years old. More information about the audio files can be found in Annex A, Section (b.1), Table B1.

Parallel to the subjective testing, the transparency of the message embedding is evaluated by computing the ODG (Objective Difference Grade) using the open source software tool EAQUAL (Evaluating of Audio QUALity, [EAQ02]). The ODG is a perceptual quality measure comparable to the SDG (Subjective Difference Grade), and is calculated on the same scale as the SDG, ranging from 0 (imperceptible) to -4 (very annoying). However the ODG is an objective measure rather than a subjective grading. According to [Thi99], the ODG is the only output of a measurement method which is considered to be directly verifiable against listening test data derived from codec comparison tests according to ITU-R Rec. BS.1116. To compare two signals with EAQUAL they both have to be resampled to 44.1 kHz, 16 bit. As both marked and unmarked files are resampled in the same way it can be assumed that the adulteration of the ODG values due to the resampling is negligible.

(b.3) Test results

In this section the test results for the speech steganography and steganalysis for the VoIP application are presented. The results can be categorised into four blocks: subjective transparency evaluation, objective transparency evaluation, steganalysis and the profile $P_{E\text{-}Long\_Time}$.

(b.3.1) Subjective tests, in combination with online tests, achieved a detection rate of less than 12% (Table 11). With an increased payload-factor the detection rate also increased. With a payload of 2 bits per packet the Positive Detection Rate (PDR = number of detected stego-files / number of all stego-files) is less than 6%, whereas with a payload of 160 Bits per packet the PDR increased to 67%. Offline-tests confirm the results reached during the online-tests. The PDR for these tests is even higher (about 49%), due to the reduced influence of the VoIP transmission (re-sampling, amplification and transmission delays are negligible in this case). Both, online- and offline-tests confirm the first test hypothesis that a subjective detection of the steganographic message at a payload-factor of 1% is not possible with a high reliability[11].

Test	***Online-Test***	***Offline-Test***

[10] A time server running on workstation C is used to synchronise the time on all three computers for the purpose of testing and logging.

[11] In this case the payload is very small. To transmit a message of 1 KByte 1:22 minutes of audio communication would be necessary, for a 1 MByte file it would require more then 23 hours.

Avg. PDR	11.7 %	49.0 %
Min. PDR	5.6 %	6.7 %
Max. PDR	67.0 %	76.1 %

Table 11: Results of subjective tests.

The test results of the objective transparency tests for 14 randomly selected audio files are shown in Figure B1, Annex A, Section (b.1). An ODG value better (greater) than -1 means that the change in the signal is almost imperceptible. All measured ODG values are better than the threshold of -1, even in the most extreme cases (payload-factor of 75% or 100%). The results of the ODG evaluation are therefore confirming the second test hypothesis that only a small acoustical modification occurs during the embedding of the steganographic message.

In [VDHK06], it was demonstrated how transparency and content in which is embedded are related for a fixed capacity. If the chosen test set is classified in speech and music it can be shown that for the same transparency the capacity for music files is significantly higher than that for speech files. For our test set we achieved an average increase by factor of up to 100. In Table B3, Annex A, Section (b.1) the context dependency of the subjective steganographic detection in the online test for six different embedding strengths and two different classes of contents (music and speech) is shown. The results from this test illustrate how the perceptibility of the watermark increases with increasing embedding strength, and how the steganographic message is more often successfully detected when embedded in speech signals than in music. Table B4 shows similar results for the context dependency of the perceptual steganographic detection such as those presented for the online tests. However, in the case of the offline tests the overall detection rates are significantly higher than in the online tests, likely due to the fact that in the offline tests the audio signal could be examined more than once by the test person. Table B5, Table B6 and Figure B9 show the results for the objective transparency evaluations for the selected test files and different embedding strengths. The results indicate that increasing the embedding strength results in audio signals which have a larger perceptual distance from the original signal. In the case of speech signals, the largest difference seems to occur with an embedding strength of 75% instead at 100%. This fact is so far unexplainable, and should be verified with a larger test set. If Tables B5 and B6 are compared, the results show that the average |ODG| value for an embedding in music with 100% embedding strength is still smaller than the average |ODG| value for an embedding into speech signals with an embedding strength of 1%. Therefore, from these preliminary results it can be inferred that, for equal transparency, music signals have a higher capacity than speech signals. These findings have to be evaluated in more detail in future work.

(b.3.2) In the evaluation of the statistical properties of the audio, the 13 attacks of the steganalyzer framework were run on the previously by random selected 14 audio signals. In most of these cases the curves of the different payload-factors are nearly identical. From these 13 attacks, seven (Median, LSB_Flipping, Covarianz, Varianz, Average, LSB_Rate and Entropy) are visualised in Figures B2-B8 in Annex A, Section (b.1). These seven attacks were the cases where the results differ most noticeably from the original audio signal. For the remaining six attacks, no visual differences between the original and the marked files are perceptible. In each of these figures the two most significant files are shown, since for the other audio signals no significant differences are noticeable.
Also visible in these figures is the fact that from the point of time when the embedding begins the curves of some of the statistical properties slightly differ from the original. However, the differences in these cases are marginal, and can only be detected reliably if the beginning of

the embedding is known to the observer. It can therefore be concluded that the second test hypothesis is also be confirmed.

(b.3.3) Evaluation of steganography and steganalysis on speech signals was done with the long time profile $P_{E\text{-}Long\_Time}$. During the 240 hours of testing no crashes or constraints in the operation occurred, confirming the hypothesis stated above that the transmission of the steganographic messages was reliable ($\eta = 0\%$).

Conclusion of the Test Results
Tests have demonstrated that VoIP communication can be practically used for steganographic applications. All test hypotheses introduced in Section (b.1) have been approved. The detection of the embedded message by applying statistical methods, the perception by human hearers and the perception by objective measures all showed that detection of the embedded steganographic message was not reliably possible. Furthermore, the tested software satisfied the long time profile ($P_{E\text{-}Long\_Time}$) with an error rate $\eta=0\%$, demonstrating the reliability of the tool in the chosen laboratory environment.

4 Accomplishments/New Findings

This section introduces new findings discovered during the evaluation of digital audio watermarks and design of distributed parallel attacking (Section a.1). In Section b.1 new findings for the speech steganography and steganalysis for the VoIP scenario are introduced. In (c.1) general ideas and an overview of future work is presented.

(a.1) New findings for the evaluation of digital audio watermarks

In this subsection, new findings observed during the implementation and evaluation of digital audio watermarking algorithms and the design of distributed parallel attacks are described. As described in [LDSV05], single attacks and audio content dependency has been identified as contributing factors to watermark embedding. If the parameters for the embedding and attacking process are the same, it is noted that different audio content has an affect on the transparency of the received audio signal. The experimental evaluation results can be summarized as follows:

- The transparency of audio watermarks showed a high dependency on the characteristics of the audio material. It was shown that single instruments are, in most cases, more affected than the other audio test files from SQAM [LDSV05].
- The audio context dependencies for the five evaluated watermarking algorithms identified that for two watermarking algorithms (LSB and Publimark) no audio context dependency could be detected. The three other watermarking algorithms evaluated (Spread Spectrum, VAWW, 2A2W) have a discernable audio context dependency, as described in subsection (a.5).
- The transparency of the embedding/attacking process $ODG(S_{EA},S_E)$ is, in 7.23% percent of all tested watermarking algorithms and all tested audio files, better than $ODG(S_E,S)$. More details about the quality improvement after attacking is shown in Annex A, Section (a.3), Table A43.

(b.2) New findings for the speech steganography and steganalysis

In this subsection, the new finding for speech steganography and steganalysis for the VoIP application is introduced.

- During the long time tests in our laboratory, expected transmission errors did not affect the transmitted message. There were no collisions which destroyed the transmitted secret message.

- The transparency of the developed steganographic embedding function is very good for low capacities, and in most cases not audible for the receiver (or an attacker). We have also shown that music data can be more transparently used that speech data.
- It is very hard to detect an embedded message (steganalysis) by computing the defined 13 statistical properties.

(c.1) General ideas and future work

(c.1.1) General new ideas:

Embedding a text message into an audio signal is only one possibility for transmitting secret information. In our future work, we propose to embed active handwriting into an audio signal and transmit this information to the receiver. The general idea behind this concept is that the statistical difference between text and an audio signal is higher that between handwriting and an audio signal. We estimate that the steganalysis required to detect an embedded handwriting signal is much more complicated than required to detect an embedded text message. The receiver will therefore not read a text message, but will rather see what the sender is actually writing as the secret information.

To evaluate this principle, two new Cintic displays have been purchased. It was observed that the statistical allocation is Gaussian for handwriting, which is similar to the speech signal. During the project 4 different new approaches to this problem could be identified which are described below:

- Two communication partners A and B are talking over the Internet using a VoIP application. In our old scenario they would use a simple text message embedded into the speech signal to transit secret information. The statistical characteristic of the text is completely different than that of the audio signal. We propose to investigate what happens when A and B use their handwriting as the embedded message into the speech signal. To detect such secret information is much more complicated than a hidden text message.
- The second approach is similar to the original VoIP application, which embeds a secret text message into the audio speech signal. However rather than the speech signal we will use the handwriting signal, which is transmitted via a HoIP (Handwriting over IP) stream over the internet. The secret text message is then embedded into the handwriting signal. Here, we estimate that the detection of the embedded information could be done by a statistical analysis.
- The third approach is the opposite of the first one described in this section. Here, the handwriting signal is transmitted in a VoIP stream over the internet and the secret information is the voice of A and/or B, which is embedded into the handwriting signal. Here, we estimate that the detection of embedded secret information is much more complicated, because of the same statistical properties of both signals (handwriting and audio).
- The last approach is a multiplexing of the aforementioned approaches. The communication partners A and B embed either their handwriting or a text message. The special application multiplexes these signals, such that the embedding function embeds handwriting or text into the audio signal. We estimate that detection of the secret information by an attacker will be much more complicated.

We believe that for all of these approaches there exists an urgent need for research. Furthermore, the developed framework can be adopted for other streaming protocols and applications, such as future applications of e-learning or video conferencing for example. For discussions about new media streaming applications at a German conference on e-learning in Leipzig (LIT05), see http://www.informatik.uni-trier.de/~ley/db/conf/lit/lit2005.html.

Future work for Audio Watermark Evaluation:
As mentioned in Section a.3 (Evaluation of Transparency), transparency enhancing methods such as the modification of attacks using psychoacoustic modelling provide better results when used for testing. Further results on this topic will be presented in an upcoming SPIE paper ([KDL06]).
The Audio WET system is now operational, and can be used by registered users. Future work may include enhancement of the Audio WET with new watermarking algorithms, new profiles and new attacks. Furthermore, the existing watermarking algorithms can and should be improved.

Future work for steganography and steganalysis for speech:
For VoIP steganography, the focus was set only to the active scenario, which was implemented and evaluated. One future work should be the implementation of VoIP scenario B (passive steganography), which includes the passive attacker scenario [DHH05], the usage of a larger test set and a different parameterization (e.g. using window sizes different from 1024 samples) for steganalysis. During the project a tool (library) providing 11 distance measures (e.g. Hamming distance, Canberra distance, Signal-to-Noise ratio) has been designed and implemented to be used for the steganalyzer framework ([SVD05]). Based on correlation coefficients, its planned use is for the comparison of feature vectors (i.e. output of the steganalyzer). In future work the impact of this approach on steganalysis has to be evaluated. As already mentioned in Section (b.3.1), the performance of the steganographic algorithm seems to depend on the content of the cover in which information is embedded (i.e. embedding in music seems to be more transparent than in speech). This has to be proven through the employment of a larger test set. Other future work includes the attachment and embedding of secret messages into GSM codecs (as described in [GOP03b] and [GOP04]) and improved synchronization (splitting the Beginning of Message (BOM) and End of Message (EOM) markers over more than one VoIP packet).

5 Related Work

This section takes a closer look to results of related work of other researches.
The evaluation of digital watermarking algorithms is a new research field. There are many attacking and evaluating tools available (StirMark[12], Checkmark[13], Certimark[14], Optimark[15], WET[16], OpenWatermark[17], etc.), but the focus is mostly on images. Purdue University (research group of Prof. Edward Delp) provides the Image WET system[18], which was the base to design and develop the Audio WET system.
Nasir Memon et. al. described a steganalyzer based on image quality metrics [AMS03]. Basically, the main idea to detect steganography by measuring distances between stego and blurred stego images, as well as between cover and blurred cover images. It is assumed that the differences between the stego files and cover files are not equal. After the determination of an optimal threshold, it is possible to create a blind distinction between stego and cover files. Characteristics of this measuring method can be taken into account to even provide indications on a per algorithm basis. It should be considered if this technique can be transferred to audio data. Blurring in the audio domain is equivalent to passing the samples

[12] http://amsl-smb.cs.uni-magdeburg.de
[13] http://watermarking.unige.ch/Checkmark/
[14] http://www.certimark.org/
[15] http://www.watermarkingworld.org/optimark/
[16] http://www.datahiding.org/ and http://audio-wet.cs.uni-magdeburg.de/wet/
[17] http://www.openwatermark.org/
[18] http://www.datahiding.org

through a low pass filter, cutting off high frequencies. This can be computed with the attacks $A_{RCLowPass}$ or $A_{FFT\_HLPassQuick}$ from SMBA.
In addition, the approach of Roy Patterson and his colleagues at the Center for the Neural Basis of Hearing in the Physiology Department of the University of Cambridge was taken into consideration for our own research. Patterson introduced an Auditory Image Model (AIM) to construct a 2-dimensional presentation from an audio file. This time-domain model is meant "[...] to simulate the auditory images we hear when presented with complex sounds like music, speech, bird songs, engines, etc." [AIM04]. Perhaps this model can help to further improve the stego attack techniques.
In [GOP03] steganographic techniques are described which embed hidden messages in GSM-coded audio data, which is patented and therefore not implemented in our framework.

6 Personnel Supported

Prof. Jana Dittmann (jana.dittmann@iti.cs.uni-magdeburg.de) – head of the research group

Claus Vielhauer (claus.vielhauer@iti.cs.uni-magdeburg.de) – researcher in the field of biometric and handwriting, benchmarking and evaluation, distance measures for clustering

Andreas Lang (andreas.lang@iti.cs.uni-magdeburg.de) – researcher in the field of digital audio watermarking and evaluation

Thomas Vogel (thomas.vogel@iti.cs.uni-magdeburg.de) – researcher in the field of illustration watermark (annotation watermarking for image and audio data)

Tobias Scheidat (tobias.scheidat@iti.cs.uni-mageburg.de) – researcher in the field of biometrics, expert in the field of benchmarking applications and distance measures, software design for web applications

Sandra Gebbensleben (sandra.gebbensleben@iti.cs.uni-magdeburg.de) – researcher in the field of audio guides and audio watermarking

Christian Krätzer (kraetzer@iti.cs.uni-magdeburg.de) – phd student in the field of steganography and steganalysis for audio data, transparency evaluation

Danny Hesse (danny.hesse@iti.cs.uni-magdeburg.de) – researcher in the field of steganography and steganalysis for audio data

Sascha Schimke (sschimke@iti.cs.uni-magdeburg.de) – phd student in the field of biometric, distance measures for data clustering

Reyk Hillert (reyk.hillert@student.uni-magdeburg.de) – student, master thesis for task (b) steganography and steganalysis in VoIP (student finalized the master thesis with in the project)

Stefan Sokoll (stefan.sokoll@student.uni-magdeburg.de) – student working for task (b), implementation of new functions for the steganalyzer used for steganalysis for VoIP

Gerald Emberger (Gerald.emberger@student.uni-magdeburg.de) – student working for task (b), implementation of modules for the steganalyzer

Michael Biermann (michael.biermann@student.uni-magdeburg.de) – student working for task (b), implementation of distance measure module used for the steganalyzer

Christian Zeitz (Christian.zeitz@student.uni-magdeburg.de) – student working for task (b), implementation of distance measure module used for the steganalyzer

Milen Touchev (milen.touchev@student.uni-magdeburg.de) – student working for task (a), evaluation of test results for the evaluation test of digital watermarking algorithms

Torsten Bölke (tboelke@iti.cs.uni-magdeburg.de) – student working for task (a), implementation of a watermarking algorithm

7 Publications

[DHH05] Jana Dittmann, Danny Hesse, Reyk Hillert; *Steganography and steganalysis in voice-over IP scenarios: operational aspects and first experiences with a new steganalysis tool set*; In: Edward J. Delp III, Ping Wah Wong (Eds.): Security, Steganography, and Watermarking of Multimedia Contents VII. Proceedings of SPIE, San Jose, CA, 2005, pp. 607-618.

[VDHK06] Thomas Vogel, Jana Dittmann, Reyk Hillert, Christian Krätzer; Design und Evaluierung von Steganographie für Voice-over-IP : To appear on Sicherheit 2006 GI FB Sicherheit, Magdeburg, 20-22 February 2006

[DHH05] Jana Dittmann, Danny Hesse, Reyk Hillert, Steganography and steganalysis in voice over IP scenarios : operational aspects and first experiences with a new steganalysis tool set : In: Delp, Edward J. (Hrsg.) ; Wong, Ping W. (Hrsg.): Security, steganography, and watermarking of multimedia contents VII (Electronic imaging science and technology San Jose, California, USA, 17-20 January 2005); Bellingham, Wash. : SPIE, 2005, pp. 607 - 618, ISBN 0-8194-5654-3, 2005

[DiHe04] Jana Dittmann, Danny Hesse; *Network Based Intrusion Detection to Detect Steganographic Communications Channels - on the Example of Audio Data*; Multimedia Signal Processing; IEEE 6th Workshop on MSP Multimedia Signal Processing, Sep. 29th - Oct. 1st 2004, Siena, Italy, ISBN 0-7803-8579-9, 2004.

[DKL05] Dittmann, J., Kraetzer, Ch. and Lang A., *Attack tuning - Attack Transparency Models and their Impact to Geometric Attacks*, 1st Wavila Challence, Barcelona, 8th-9th June 2005

[EC05] ECRPT NoE, http://www.ecrypt.eu.org/, 2005

[KDL06] Christian Kraetzer, Jana Dittmann, Andreas Lang, Transparency benchmarking on audio watermarks and steganography, to appear in SPIE conference, at the Security, Steganography, and Watermarking of Multimedia Contents VIII,

IS&T/SPIE Symposium on Electronic Imaging, 15-19th January, 2006, San Jose, , USA, 2006

[LD04] Andreas Lang, Jana Dittmann: StirMark and profiles: from high end up to preview scenarios, Virtual Goods 2004, 28. - 29.05., Ilmenau, Germany, to appear in http://virtualgoods.tu-ilmenau.de/2004/, 2004

[LD06] Andreas Lang, Jana Dittmann, Profiles for Evaluation - the Usage of Audio WET, to appear in SPIE conference, at the Security, Steganography, and Watermarking of Multimedia Contents VIII, IS&T/SPIE Symposium on Electronic Imaging, 15-19th January, 2006, San Jose, , USA, 2006

[LDLD05] Andreas Lang, Jana Dittmann, Eugene T. Lin, Edward J. Delp, Application Oriented Audio Watermark Benchmark Service, In: Edward J. Delp III, Ping Wah Wong (Eds.): Security, Steganography, and Watermarking of Multimedia Contents VII. Proceedings of SPIE, San Jose, CA, 2005

[LDSV05] Andreas Lang, Jana Dittmann, Ryan Spring, Claus Vielhauer; Audio Watermark Attacks: From Single to Profile Attacks; ACM Multimedia '05 M&S, 1-2 August New York, ISBN 1-59593-032-9, 2005, pp. 39-50, 2005

[LDLD05] Andreas Lang, Jana Dittmann, Eugene T. Lin, Edward J. Delp, *Application Oriented Audio Watermark Benchmark Service*, In: Edward J. Delp III, Ping Wah Wong (Eds.): Security, Steganography, and Watermarking of Multimedia Contents VII. Proceedings of SPIE, San Jose, CA, 2005

[LDSV05] Andreas Lang, Jana Dittmann, Ryan Spring, Claus Vielhauer; *Audio Watermark Attacks: From Single to Profile Attacks*; ACM Multimedia '05 M&S, 1-2 August New York, ISBN 1-59593-032-9, 2005

[SVD05] Tobias Scheidat, Claus Vielhauer, Jana Dittmann, Distance-level fusion strategies for online signature verification, In: Institute of Electrical and Electronics Engineers, IEEE (Veranst.): Multimedia and expo, ICME 2005 (IEEE international conference Amsterdam July 6th - 8th 2005 ; ISBN 0-7803-9332-5, 2005

[VD05] Thomas Vogel, Jana Dittmann; Illustration Watermarking: An object based approach for digital images : In: Delp, Edward J. (Hrsg.) ; Wong, Ping W. (Hrsg.): Security, steganography, and watermarking of multimedia contents VII (Electronic imaging science and technology San Jose, California, USA, 17-20 January 2005) ; Bellingham, Wash. : SPIE, pp. 578 589, ISBN 0-8194-5654-3, 2005

[VDHK06] Thomas Vogel, Jana Dittmann, Reyk Hillert, Christian Krätzer: Design und Evaluierung von Steganographie für Voice-over-IP, To appear in conference „GI Sicherheit 2006", February 20-20, Magdeburg, Germany, 2006

8 Interactions/Transitions

None

9 New discoveries, inventions, or patent disclosures

None

10 Honors/Awards

None

11 References

[AIM04] Auditory Image Model (AIM), Website: http://www.mrc-cbu.cam.ac.uk/cnbh/web2002/framesets/AIMframeset.htm, 2004.

[AMS03] Ismail Avcıbas, Nasir Memon, Bülent Sankur, *Steganalysis based on Image Quality Metrics*, IEEE Transactions on Image Processing, Vol. 12, No. 2, February 2003

[BTH96] Boney, L., Tewfik, A. H., and Hamdy, K. N., *Digital watermarks for audio signal*, International Conference on Multimedia Computing and Systems, Hiroshima, Japan, pp. 473-480, 1996

[CKL96] Cox, I.J., Kilian, J., Leigton, F.T., and Shamoon, T., *Secure Spread Spectrum Watermarking for Multimedia*, IEEE Trans. Image Processing, vol. 6, pp. 1673-1687, 1996

[CKS01] Cvejic, N., Keskinarkaus, A. and Seppanen, T., *Audio watermarking using m-sequences and temporal masking*, IEEE Workshops on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York, pp. 227-230, 2001

[CVE04] Nedeljko Cvejic, *Algorithms for Audio Watermarking and Steganography*, Department of Electrical and Information Engineering, Information Processing Laboratory, University of Oulu, 2004

[DRA01] Dugad, R., Ratakonda, K. and Ahuja, N., "*A New Wavelet-Based Scheme for Watermarking Images*", Department of Electrical and Computer Engineering, Beckmann Institute, University of Illinois, Urbana, IL 61801, 2001

[EAQ02] Alexander Lerch, zplane.development, *EAQUAL - Evaluation of Audio Quality*, Version: 0.1.3alpha, http://www.mp3-tech.org/programmer/misc.html, 2002

[EBTG03] Eggers, J. J., Bäuml, R., Tzschoppe, R. and Girod, B., *Scalar Costa scheme for information embedding*, IEEE Trans. on Signal Processing, 2003

[GFD02] Guillon, P., Furon, T. and Duhamel, P., *Applied public-key steganography*, in Proc. SPIE, San Jose, CA, USA, 2002

[GLF+93] Garofolo, Lamel, Fisher, Fiscus, Pallett, Dahlgren, *DARPA, TIMIT, Acoustic-Phonetic Continuous Speech Corpus*, CD-ROM, NIST Speech Disc 1-1.1 U.S. Department of Commerce Technology Administration National Institute of Standards and technology, CD-Rom October 1990, Documentation 1993

[GLO97] Glover, I.A. & Grant, P.M., *Digital Communications*, Prentice Hall 1997

[GOP03] K. Gopalan: Audio Steganography by Amplitude or Phase Modification, Proc. Of 15th Annual Symposium on Electronic Imaging -- Security, Steganography, and Watermarking of Multimedia Contents V, San Jose, 2003.

[GOP03b] Goplan, Wenndt, Noga: *Covert Speech Communication Via Cover Speech by Tone Insertion*, Proc. Of the 2003 IEEE Aerospcae Conference, Big Sky, Department of Engineering, Purdue University Calument, Hammond, Multi-Sensor Exploration Branch, ARFL/IFEC, Rome, 2004

[GOP04] Goplan, Wenndt: *Audio Steganography for Covert Data Transmission by Imperceptible Tone Insertion*, Department of Engineering, Purdue University Calument, Hammond, Multi-Sensor Exploration Branch, ARFL/IFEC, Rome, 2004

[Gue05] Guelvouit, G. Le., *Trellis-coded quantization for public-key watermarking*, accepted for IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2005

[IMYK99] Inoue, H., Miyazaki, A., Yamamoto, A., Katsura, T., *A Digital Watermarking Technique Based on the Wavelet Transform and Its Robustness on Image Compression and Transformation*, IEICE Trans. Fundamentals, vol. E82-A, no. 1, 1999

[King99] Y. Duan, I. King, *A Short Summary of Digital Watermarking Techniques for Multimedia Data*, Department of Computer Science & Engineering, The Chinese University of Hong Kong. Shatin, N. T., Hong Kong, China, 1999

[Kim00] Kim, H., *Stochastic model based audio watermark and whitening filter for improved detection*, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, pp. 1971-1974, 2000

[Kim03] Kim, H.J., *Audio Watermarking Techniques*, Pacific Rim Workshop on Digital Steganography, Kyushu Institute of Technology, Kitakyushu, Japan, 2003

[KM01] Kirovski, D., and Malvar, H., *Robust spreadspectrum audio watermarking*, IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UT, pp. 1345-1348, 2001

[KRA05] Kraetzer, Ch., *Improving Attack Transparency of Audio Watermarks by Using Psychoacoustic Methods*, Diploma Thesis, Research Group Multimedia and Security, Department of Computer Science, Otto-von-Guericke-Universität Magdeburg, P.O. Box 4120, 39016 Magdeburg, Germany, April 30th, 2005

[LH00] Lee, S.K., and Ho, Y.S., *Digital audio watermarking in the cepstrum domain*, IEEE Transactions on Consumer Electronics, vol. 46, no. 3, pp. 744-750, 2000

[LDS03] Lang, A., Dittmann, J. and Steinebach, M., *Psycho-akustische Modelle für StirMark Bechmark - Modelle zur Transparenzevaluierung*, Rüdiger Grimm; Hubert B. Keller; Kai Rannenberg (eds.), Sicherheit - Schutz und Zuverlässigkeit, Informatik 2003 - Mit Sicherheit Informatik, pages 399-410, October 2003, Frankfurt/Main, ISBN 3-88579-365-2, 2003

[LS04] Andreas Lang, Ryan Spring, *StirMark Benchmark for Audio - Attack Description*, internal report, 2004

[Popa98] Popa, R., *An Analysis of Steganographic Techniques*, The "Politehnica" University of Timisoara Faculty of Automatics and Computers Department of Computer Science and Software Engineering, 1998

[PubHt] Publimark, http://gleguelv.free.fr/soft/publimark/, 2005

[Sha93] Shapiro, J.M., *Embedded image coding using zerotrees of wavelet coefficients*, IEEE Trans. Signal Processing, vol. 41, no.12, pp. 3445-3462, 1993

[SHK02] Seok, J., Hong, J., and Kim, J., *A novel audio watermarking algorithm for copyright protection of digital audio*, ETRI Journal, vol. 24, pp. 181-189, 2002

[SMBA] StirMark Benchmark for Audio, http://amsl-smb.cs.uni-magdeburg.de, 2005

[SQAM] SQAM - Sound Quality Assessment Material, http://www.tnt.uni-hannover.de/project/mpeg/audio/sqam/, 2005

[SZTB98] Swanson, M., Zhu, B., Tewfik, A., and Boney, L., *Robust audio watermarking using perceptual masking*, Signal Processing, vol. 66, pp. 337- 355, 1998

[Thi99] Thiede, T., Perceptual Audio Quality Assessment using a Non-Linear FilterBank, PhD Dissertation, Technische Universität Berlin, Fachbereich Elektrotechnik, 1999