IST-2002-507932

ECRYPT

European Network of Excellence in Cryptology

Network of Excellence
Information Society Technologies

D.WVL.21

Final Report on Watermarking Benchmarking

Due date of deliverable: 31. July 2008
Actual submission date: 31. July 2008

Start date of project: 1 February 2004
Duration: 4.5 years

Lead contractor: Katholieke Universiteit Leuven (KUL)

Revision 1.0

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th>PU</th>
<th>PP</th>
<th>RE</th>
<th>CO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restricted to other programme participants (including the Commission services)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restricted to a group specified by the consortium (including the Commission services)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confidential, only for members of the consortium (including the Commission services)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Final Report on Watermarking Benchmarking

Editor
Andreas Lang (GAUSS),
Jana Dittann (GAUSS),
Christian Krätzer (GAUSS)

Contributors
Christian Krätzer (GAUSS),
Elke Franz (GAUSS),
Jana Dittmann (GAUSS),
Andreas lang (GAUSS)

Revision 1.0

The work described in this report has in part been supported by the Commission of the European Communities through the IST program under contract IST-2002-507932. The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.
# Contents

1 Introduction 1

2 The Connection between Watermarking Benchmarking and Steganalysis 3
   2.1 Identification of the Connection between Watermarking Benchmarking and Steganalysis 3
   2.2 A Selection of Work Performed by WAVILA WP3 in Audio Watermark Benchmarking and Steganalysis 5
      2.2.1 Application Specific Benchmarking for Watermarking and Steganography 5
      2.2.2 Detectability Evaluations on Exemplarily Chosen Watermarking and Steganography Algorithms using a Steganalysis Tool Set 6
      2.2.3 Presentation and Visualisation of Benchmarking Results and the Impact of Applied Benchmarking to the Benchmarking Tools 6
   2.3 A Brief Summary of the Benchmarking Problem for Watermarking and Steganography 7

3 Digital Watermark Benchmarking on Example of Audio 9
   3.1 Theoretical Framework 9
      3.1.1 Terminology of Watermark Properties 10
         3.1.1.1 Basic Definition 10
         3.1.1.2 Instance of a Watermarking Scheme 11
         3.1.1.3 Embedding Function 12
         3.1.1.4 Detection and Retrieval Function 15
         3.1.1.5 Watermark Attacking Functions 20
         3.1.1.6 Other Properties – Invertibility, Verification, Security 26
         3.1.1.7 Realization of Security Measurements 29
      3.1.2 Quality Dependent Benchmarking 31
      3.1.3 Watermarking Life-Cycle Phases 34
3.1.4 Benchmarking from the Application Point of View ...................... 36
3.1.5 Benchmarking Metrics for the Profile Based Approach .................. 38
  3.1.5.1 Definition of Basic Profiles ...................................... 40
  3.1.5.2 Definition of Extended Profiles .................................... 44
  3.1.5.3 Definition of Application Profiles ................................. 49
3.1.6 Evaluation Methodology .................................................. 63
  3.1.6.1 Evaluation Methodology based on Watermarking Parameters ....... 63
  3.1.6.2 Evaluation Methodology Based on Profiles ......................... 66
3.1.7 Audio Data Test Set: Formalization and Example Test Sets ............. 68
3.2 Practical Framework .......................................................... 72
  3.2.1 Test Goals ............................................................... 72
  3.2.2 Selected Watermarking Schemes for Evaluation ....................... 73
  3.2.3 Test Scenario ............................................................ 78
3.3 Test Results of Profile Based Evaluation of Digital Audio Watermark Schemes with the Application Profile Perceptual Hashing .................... 81
3.4 Conclusion and Outlook ...................................................... 90

4 Steganalytical Results for a Steganographic Algorithm ........................ 91
  4.1 Introduction .................................................................. 91
  4.2 Steganographic Algorithms ................................................. 92
  4.2.1 Embedding Considering Adjacent Pixels (ECAP) ..................... 92
  4.2.2 $\pm 1$ Steganography or LSB Matching (LSBM) ..................... 93
  4.2.3 Stochastic Modulation (StM) .......................................... 94
  4.2.4 Perturbed Quantization (PQ) ............................................ 94
  4.3 Steganalytical Methods used in the Tests ................................ 95
  4.3.1 Rationale for Selection ................................................. 95
  4.3.2 Selected Classifiers ..................................................... 95
  4.3.3 Problem: Size of the Test Set ....................................... 96
  4.4 Practical Results ............................................................ 97
  4.4.1 Test Sets and Test Procedure ...................................... 97
  4.4.2 Preliminary Considerations ........................................... 98
  4.4.3 Evaluating the Approaches for Reducing the Effort ................ 99
    4.4.3.1 Reduced Image Size ............................................ 99
4.4.3.2 Reduced Feature Vectors ........................................... 101
4.4.3.3 Conclusions ......................................................... 102
4.4.4 Compare ECAP, LSBM, and StM .................................... 102
4.4.5 Compare ECAP, LSBM, and PQ .................................... 104
4.5 Conclusion and Outlook .................................................. 106

5 Summary ........................................................................... 107

6 References ........................................................................ 111
ECRYPT — European NoE in Cryptology
Chapter 1

Introduction

This document is the final report on watermarking benchmarking, of WAVILA WP3, which is a work package within the WAVILA virtual lab of the ECRYPT Network of Excellence in Cryptology. The research results done within this project are summarized and presented for watermarking benchmarking and steganalysis. In this document, the focus is set on three main aspects. Firstly, a short summary of the work done within this project is given to give a brief overview. Application scenarios are focus of the evaluation of digital audio watermarking schemes and steganographic techniques. Furthermore, the detectability evaluation of watermarking and steganographic techniques are in focus. One main part of this final report is the formalization and definition of fundamental watermarking parameters with their normalized measurements. The exemplary selected application scenario of perceptual hashing is used to show the usability of the theoretical definitions by evaluating exemplary selected digital audio watermarking schemes. Note, the final report on forensic tracking techniques of WP4 [Ope08] introduces and discusses different algorithms on fingerprinting and perceptual hashing for image and video data. This WP3 final report includes also project summaries of steganalytical results on exemplary selected steganographic algorithms for images.

This report is structured as follows. In chapter 2 a summary of the work done in WAVILA WP3 is given. Therefore, in section 2.1 the connection between digital watermarking benchmarking and steganalysis is discussed whereby in section 2.2 exemplary selected work done in the area of digital audio watermarking benchmarking and steganalysis are summarized and discussed. The chapter 2 closes in section 2.3 with a summary of benchmarking problems for digital watermarking and steganography.

In chapter 3 is digital audio watermarking benchmarking defined, formalized and presented. In section 3.1 is the theoretical framework introduced, which defines on one hand the fundamental watermarking parameters and on the other hand evaluation profiles to support and simplify the evaluation process itself. The practical framework, which shows the usage of the theoretical framework on the example of perceptual hashing, is shown in section 3.2. The evaluation test results on the example of application oriented evaluation with the application profile perceptual hashing are presented in section 3.3. The chapter closes with a brief summary and outlook in section 3.4.

The chapter 4 focuses on steganalytical evaluation results of steganographic algorithms. Section 4.1 gives a short introduction to the topic. Exemplary selected steganographic algorithms are briefly summarized and introduced in section 4.2. The steganalytical methods, useable for
identification and classification of marked objects with the steganographic algorithm Embedding Considering Adjacent Pixels, are described in section 4.3. The practical evaluation of the exemplary selected steganographic algorithms is described in section 4.4. In section 4.5, the chapter closes with a summary and an outlook. Chapter 5 closes the report with a conclusion.
Chapter 2

The Connection between Watermarking Benchmarking and Steganalysis

During the 4.5 years work in WAVILA WP3 many important results have been achieved in the domain of watermarking benchmarking and steganalysis. In the following sections 2.1 to 2.3, first an introduction into the connection between watermarking benchmarking and steganalysis is given (in section 2.1) to show how one of these domains might gain benefit from improvements in the other and vice versa. Second, an overview, based on exemplarily selected publications, of the work of WP3 is given for the areas of audio watermarking benchmarking and audio steganalysis (see section 2.2). In section 2.3 a brief summary of the benchmarking problem for watermarking and steganography is given.

2.1 Identification of the Connection between Watermarking Benchmarking and Steganalysis

Using the classification introduced by Katzenbeisser and Petitcolas in [KP00], digital watermarking (a.k.a. watermarking) and steganography are two closely related sub-categories of information hiding. Both are commonly joined in the term data hiding [Fri98a]. By quoting Cox’s [CMB+08] definitions on watermarking and steganography we can state that:

Watermarking is the practice of imperceptibly\(^1\) altering a digital object to embed a message about that object.

And:

Steganography is the practice of undetectably altering a digital object to embed a secret message.

\(^1\)Note: not all researchers consider imperceptibility a defining characteristic of digital watermarking - some application scenarios call for perceptible watermark embedding, which leads to the field of perceptible watermarking
Therefore in both cases additional payload is embedded into an digital object called the cover. In watermarking the payload is somehow correlated to the object (e.g. ownership information, metadata, integrity verification information, annotations, copy control information, etc) while in steganography the sole purpose of the embedding is either hidden communication or hidden data storage. In watermarking the detector who detects/retrieves the watermark from the marked object will make use of both data while in steganography the detector might even discard the stego object after the retrieval of the message.

Based on the different goals for watermarking and steganography there also exist differences in the evaluation of the schemes. In watermarking a benchmarking of algorithms is performed (ideally) a priori to the usage of these algorithms to allow for a comparison of performance under well defined requirements. This is necessary to allow end users to select an algorithm which suits their application scenario. In steganography the counter science of steganalysis tries to detect on demand (sometimes even in real-time) the existence of hidden communication channels. An a priori evaluation like in watermarking benchmarking is less common but can be used to tune/improve steganalysers and/or the steganography algorithms. Also sometimes the a posteriori usage of steganalytic techniques is seen (e.g. in media forensics approaches like presented in [KODL07]).

An early visualisation used in the description of data hiding techniques is seen in figure 2.1. This figure shows on the trade-off between the characteristics capacity, transparency, and robustness the difference between watermarking and steganography as it was perceived in the late 1990s. Meanwhile new application scenarios for watermarking have emerged (fragile watermarks, just to mention one) and these are not necessarily compliant to the classification behind this visualisation. What has not changed is the fact that for steganography the non-detectability of the hidden channel is still the most important of the characteristics. Other characteristics, which might be of stronger importance in watermarking, like robustness, capacity, etc are less important in steganography. Because of this it is natural that we find in steganalysis very advanced methods for detectability/non-detectability evaluations for steganography algorithms which can also be used for similar evaluations in watermarking benchmarking.

![Figure 2.1: Trade-off between capacity, transparency, and robustness. [Fri98a]](image-url)
In general, since watermarking and steganography algorithms principally employ similar techniques for different goals, the idea of using evaluation techniques from one domain also in the other seems natural. Previous WP3 deliverables (D.WVL.10 Audio Benchmarking Tools and Steganalysis [Dit06] and D.WVL.16 Report on Watermarking Benchmarking and Steganalysis [Dit07]) already discussed this idea by showing how watermarking benchmarking tools like SMBA ([LS04]) and Audio WET ([LDLD05], [LD06b]) - both developed by ECRYPT partner GAUSS - can be used also for steganography algorithms or how the AMSL Audio Steganalysis Toolset (AAST [Dit07], also developed by GAUSS) can be used in the benchmarking of watermarking techniques.

In the following section an overview of the results achieved in ECRYPT WP3 on audio data hiding evaluations is presented. The knowledge gathered on this type of media can not be generalised without further research for other kinds of media but it can serve as an indicator for the applicability of the techniques.

2.2 A Selection of Work Performed by WAVILA WP3 in Audio Watermark Benchmarking and Steganalysis

Besides research work on “classical” watermarking benchmarking, like e.g. [LDMHJ06] which compares practical watermarking benchmarking results for selected algorithms considering robustness, transparency and capacity or [LD06c] where a transparency and complexity benchmarking of audio watermarking algorithms issues is performed, the focus in the audio domain benchmarking performed in WP3 was directed towards application specific benchmarking, comparisons in the detectability evaluation of watermarking and steganography, visualisation of benchmarking results and the improvement of benchmarking tools. A brief summary from the work in these domains is given in the following three subsections.

2.2.1 Application Specific Benchmarking for Watermarking and Steganography

The theoretical concept developed by the ECRYPT partner GAUSS and introduced in section 3 is manifested partially in evaluation tools like SMBA (Stirmark Benchmark for Audio [LS04]) or the Audio WET (Watermark Evaluation Testbed [LDLD05, LD06b]). One of the main ideas - the application specific benchmarking (or application profile based benchmarking) - is used for many different classes of potential applications for watermarking and steganographic techniques. Five of these are indicated below:

- **Application of watermarking in biometrics:** publications like [LD07] and [OLV06] evaluate digital speech watermarking and its impact to biometric speech authentication.

- **Application of watermarking in perceptual hash watermarking:** one point were the research interests in WAVILAs WP3 (Benchmarking) and WP4 (Perceptual Hashing) overlapped is the benchmarking of perceptual hashing methods. In publications like [LD08] and [LDK07a] the basic ideas of using perceptual hashes as a quality measure for watermarking as well as original and watermarked material as a test set for perceptual hash stability/collusions are developed and evaluated.
• **Using watermarking as a trigger in location based services:** in [KLD+08] the usage of watermarked background signals is briefly discussed for its application in location based services on the basis of mobile phones, exploiting their audio recording functionalities.

• **Benchmarking in annotation watermarking:** the application field of annotation watermarking allows for certain basic assumptions on the attacker behaviour. Since here the occurrence of malicious modifications (malicious attacks) is not assumed, but non-malicious modifications (like pre-processing for different display devices) should be allowed, an adjusted benchmarking procedure as shown by [VSKD08] for transparency and capacity should be performed.

• **VoIP steganography and watermarking:** a number of WP3 publications compares in detail the detectability for watermarking and steganography algorithms in the application scenario of data hiding in the audio component of VoIP streams. Exemplarily selected publications in this application field are: [KD07a], [KD06], [KDVH06], and [VDHK06]

### 2.2.2 Detectability Evaluations on Exemplarily Chosen Watermarking and Steganography Algorithms using a Steganalysis Tool Set

A considerable fraction of the publications in WP3 is dedicated to detectability evaluations on exemplarily chosen watermarking and steganography algorithms using the AMSL Audio Steganalysis Toolset (AAST). In the following some exemplarily publications are listed and their focus is highlighted: [KDL06] comparing basic transparency models for watermarking and steganography, identifying the differences in the basic requirements for this characteristic. In [KD07b] a model cross-evaluation (using classifier models generated for watermarking algorithms in the detectability evaluation for steganographic methods and vice versa) is performed. This publication shows that models generated for model based steganalysis can also be used in practise to detect/correctly classify watermarking algorithms. The impact of training on cover dependent and independent training/testing in statistical detectability evaluations in watermarking and steganography is considered e.g. in [KD08]. The cover signal specific steganalysis performed in this publication evaluates the impact of two different basic assumptions in the training/test set composition on the detection process. One of the results from the mentioned papers is the confirmation of the expectation that the practically evaluated watermarking algorithms show a higher statistical detectability when compared to the steganography algorithms. This is explained with the different design goals (for steganography non-detectability is the most important characteristic and everything else is secondary, while in practical watermarking application most often robustness and/or capacity are the most important characteristics).

### 2.2.3 Presentation and Visualisation of Benchmarking Results and the Impact of Applied Benchmarking to the Benchmarking Tools

Two further research topics in WP3 have been the communication of benchmarking results to non-experts and the improvement of benchmarking tools. Despite much more effort within
WAVILAs work was spend on this, only two exemplary publications for each of these two topics shall be mentioned here: In [Kra06] different visualisation strategies from information visualisation are discussed for their applicability in watermarking benchmarking. The goal is to give researchers some indication how to enable non-experts to choose the “best” algorithm for their application scenario by designing easy to interpret visualisations for benchmarking results. A practical usage of this idea can be seen in [LDMHJ06] where the authors visualise benchmarking results in the triangular trade-off between robustness, transparency and capacity shown in figure 2.1.

By performing actual benchmarking also new ideas how to improve the benchmarking process emerge. For example [DKL05] shows how SBMA is improved in terms of transparency of applied geometrical attacks by perceptual modelling in the attack process. From the field of steganalysis and the considerations for improvement of the AAST the publication [KD07b] shall be mentioned here as an evaluation of a functionality improvement.

2.3 A Brief Summary of the Benchmarking Problem for Watermarking and Steganography

Fair watermarking benchmarking is quite a hard to solve problem. This knowledge derived from early publications on this topic (e.g. [KP99]) still holds and the complex methodology presented in section 3 of this report illustrates very well why the already large but still increasing number of different possible applications for watermarking techniques keeps the benchmarking very difficult. For the domain of steganographic algorithms, where the application goal is much narrower, we face the basic problem that the steganographic schemes should be designed to be non-detectable. Nevertheless this basic assumption of an implemented steganographic algorithm must be evaluated in practise to detect, amongst other problems, eventually existing content dependencies (e.g. just to give some practical examples from audio steganography/steganalysis: algorithms not being able to embed into white noise or embedding into digital silence and therefore being very detectable in this case) which defy this requirement.

During the 4.5 years work of WAVILA WP3 on this topic many important results have been achieved in this domain and an corresponding number of publications reflects these results. We consider our work as an contribution to the benchmarking domain which helped in the maturing process of this field even if a large number of problems is still subject to ongoing and future work.
Chapter 3

Digital Watermark Benchmarking on Example of Audio

In this chapter the evaluation of digital watermarking algorithms is clearly defined in the theoretical framework. Thereby, the fundamental watermark properties are discussed and their normalized measurement described. Derived from it, so called profiles are defined, which can be used on one hand from watermark designer with deep inside knowledge and on the other hand from end-users with few technical knowledge about the technology of digital watermarking. This chapter contains also a practical framework, where six exemplary selected digital audio watermarking schemes are chosen for an application oriented evaluation of their embedding transparency. The application oriented evaluation is presented on the example of perceptual hashing. The presented research results of the theoretical and practical framework are described in more details in [Lan08]. Please note, the defined theoretical framework can be easily adapted on other media types like images, video or 3D.

This chapter is structured as follows: in section 3.1 the theoretical framework and in section 3.2 the practical frameworks are defined and introduced. In section 3.3 are our practical evaluation results presented and discussed. This chapter closes in section 3.4 with a summary and future work.

3.1 Theoretical Framework

In this section, the theoretical framework for the evaluation of digital watermarking schemes is defined and introduced. Thereby, the fundamental watermarking properties are with their required measurement functions in subsection 3.1.1 described. In subsection 3.1.2 is an audio quality depending evaluation introduced, whereby in subsection 3.1.3 the watermark evaluation is focused on the watermarking life-cycle phases. In subsection 3.1.5 the profile based benchmarking metric is introduced and described for selected basic-, extended- and application profiles. This section ends with an evaluation methodology in subsection 3.1.6 and a defined test set on the example of audio in subsection 3.1.7.
3.1.1 Terminology of Watermark Properties

Motivated from the different definitions of watermarking properties and watermark functions, the theoretical framework defines and formalizes the watermark properties and their measurements clearly in this section. The basic results are based on the joined work of Dittmann, Megías, Lang and Herrera-Joancomartí and also published in [Lan08]. The various types of digital audio watermarking algorithms differ in their input parameters for the embedding and detection/retrieval function. Thereby different general properties of watermarking algorithms and the embedded watermarks exist which are structured and formalized in the following [DMLHJ06, Gar02, Ace04]. The basic definitions, introduced in [DMLHJ06], defines the properties of a digital audio watermarking scheme to provide comparability of different watermarking schemes to each other by normalizing the measurements.

3.1.1.1 Basic Definition

In this subsection the theoretical basic definitions for watermark evaluation to compare different watermarking schemes are provided. Therein, we introduce the watermarking scheme, the cover and the marked object, the embedding message and the overall watermarking properties.

A *watermarking scheme* $\Omega$ can be defined as the 7-tuple given by

$$\Omega = (E, D, R, M, \mathcal{P}_E, \mathcal{P}_D, \mathcal{P}_R),$$

where $E$ is the embedding function, $D$ is the detection function, $R$ is the retrieval function, $M$ is the domain of the hidden message and $\mathcal{P}_E$, $\mathcal{P}_D$, $\mathcal{P}_R$ are, respectively, the domains for the parameter settings used for embedding, detection and retrieval.

Although more precise definitions are provided below for the different functions involved, it is worth pointing out that the detection and retrieval functions are often dependent. On one hand, some schemes only provide a method to detect whether the watermark is present in an object or not. These schemes define detection functions $D$ but no retrieval mechanisms. On the other hand, different schemes make it possible to recover an identified version of the embedded message and a retrieval $R$ function is defined. In such a case, a detection function $D$ may be defined in terms of the retrieval function. For example, the retrieved message should be identical to the embedded one (at least above some threshold) to report detection. An example of this kind of detection function defined in terms of retrieval is the spread spectrum scheme in [CKLS96].

The important properties of watermarking schemes are usually applied to assess performance, namely *robustness*, *capacity* and *transparency* [Fri98b]. Often, an improvement in one of these properties implies a decline in some of the other ones and, thus, some trade-off solution must be attained. For example, if robustness is increased by optimizing the watermark embedding parameters, then the capacity and/or transparency is often decreased. If the capacity can be increased, then in most cases the robustness or transparency decreases. The following Figure 3.1 introduces the triangle (often called magic triangle) between the three properties on two examples [Dit00]. The embedding parameters for the watermarking scheme $\Omega_A$ are tuned to provide high robustness. The price for the robustness of $\Omega_A$ is a bad transparency and a low embedding capacity. Therefore, $\Omega_A$ is located close to the robustness
corner of the triangle. Watermark $\Omega_B$ is tuned for a high transparency. The result is a low robustness and a low capacity. Therefore $\Omega_B$ is located close to the transparency corner of the triangle.

In [CAY07] the super robustness of digital watermarks is introduced, which is an extreme property of a given $\Omega$, whereby it is located at the corner of “Robustness=1”. Super robustness means, that the embedded watermark is extremely robust against specific attacks or an attack changes many parts of the signal except the marking positions itself. The result is an attacked signal, with a very worse attacking transparency (the attacked signal seems to be completely different than the marked signal) but the watermark is detect and retrievable.

![Figure 3.1: Illustration of the Trade–Off between Robustness, Transparency and Capacity](image)

If other properties of the watermark are needed, then the algorithm parameters (if possible) can be modified to locate the watermark on any point inside the triangle in Figure 3.1. The requirements of the properties depend on the application used. Remark: an algorithm with 50% transparency, 50% capacity and 50% robustness or 100% transparency, 100% capacity and 100% robustness unfortunately would be positioned in the middle of the triangle.

The following subsections define and discuss the watermarking properties and their association to the embedding, attacking and detection/retrieval functions. Their normalized measurements are introduced, which are required as internals of the profile based evaluation approach.

### 3.1.1.2 Instance of a Watermarking Scheme

With Equation (3.1) a general watermarking scheme is defined where several parameters can adopt different values, which result on the joined work [DMLHJ06]. In particular, there are embedding parameters $p_E \in \mathcal{P}_E$, detection parameters $p_D \in \mathcal{P}_D$ and retrieval parameters $p_R \in \mathcal{P}_R$. Hence, each watermarking scheme $\Omega$ may have different instances according to the values that these parameters may adopt. An instance $\Omega^*$ of the watermarking scheme $\Omega$ for
a particular value of the parameter vectors is defined as:

$$\Omega^* = (E, D, R, M, p_E, p_D, p_R),$$

(3.2)

for $p_E \in P_E$, $p_D \in P_D$ and $p_R \in P_R$.

Cover and marked object

The cover object $S$ is the original content to be marked. Here, the general term “object” is used to refer to audio signals (or if adapted to other types of media: digital images, video and any other which can be marked). Once the message is embedded into the object $S$, a marked object $S_E$ is obtained.

Watermark and message

Depending on the watermarking algorithm, the watermark message $m$ is given by the application or the user. In addition, it must be taken into account that the message $m$ and the actual embedded bits may differ. For example, redundancy may be introduced for error detection or correction [DFHJ00]. Hence, the notation $w$ to denote the watermark (or mark) which refers to the true embedded bit stream is introduced. $w$ is obtained as the result of some coding function of the message $m$. In any case, the embedding capacity of a watermarking scheme is measured according to the entropy of the original message $m$ and not the embedded mark $w$:

$$w = \text{cod}(m, p_{\text{cod}}),$$

(3.3)

where cod is some coding function and $p_{\text{cod}} \in P_{\text{cod}}$ where $P_{\text{cod}} \subseteq P_E$ are the coding parameters. These parameters may include secret or public keys for security reasons.

Classification according to the length of the transmitted message

The length of the embedded message $|m|$ determines two different classes of watermarking schemes:

- $|m| = 0$: The message $m$ is conceptually zero-bit long and the system is designed in order to detect only the presence or the absence of the watermark $w$ in the marked object $S_E$. This kind of schemes is usually referred to as zero-bit or presence watermarking schemes. Sometimes, this type of watermarking scheme is called 1-bit watermark, because a 1 denotes the presence and a 0 the absence of a watermark.

- $|m| = n > 0$: The message $m$ is a $n$-bit long stream ($m = m_1 \ldots m_n$, $n \in \mathbb{N}$, with $n = |m|$) or $M = \{0, 1\}^n$ and is modulated in $w$. This kind of schemes is usually referred to as multiple bit watermarking - or non zero-bit watermarking schemes.

3.1.1.3 Embedding Function

Given the cover object (such as an original unmarked audio signal) $S$, the watermark or mark $w$ and a vector of embedding parameters $p_E$, the marked object $S_E$ is obtained by means of
an embedding function \( E \) as follows:

\[
S_E = E(S, w, p_E) = E(S, \text{cod}(m, p_{\text{cod}}), p_E),
\]

where specific values must be provided for the coding and the embedding parameters, \( p_{\text{cod}} \) and \( p_E \in P_E \), where \( P_E \) denotes the domain for the embedding parameters.

The embedding process can usually be tuned with different parameters. In addition, it must be taken into account that several watermarking schemes require public or private (encryption) keys defined by the Kerckhoffs principle to introduce security. Those keys \( k \) belong to a key space \( \mathcal{K} \) \((k \in \mathcal{K})\) and, if present, are also a component of the vector \( p_E \) of embedding parameters. If a watermarking scheme embeds \( m \) multiple times and can be controlled by a parameter \( p_{\text{max}} \), then it is part of \( p_E \).

**Embedding Capacity:** The embedding capacity \( \text{cap}_E \) of a watermarking scheme is defined as the amount of information that is embedded into the cover object to obtain the marked object. A simple definition for a capacity measure \( \text{cap}^*_E \) would be related to the size of the embedded message, \( \text{i.e.} \ \text{cap}^*_E(\Omega, S_E) = \text{size}(m) = |m| \). In addition, capacity is often given relative to the size of the cover object:

\[
\text{cap}_{E_{\text{rel}}}(\Omega^*, S) = \frac{\text{cap}^*_E}{\text{size}(S)}.
\]

Note that such measure only takes into account the information embedded, but not the information that is retrieved. Note, also, that this measure does not consider the possibility of repeat coding, in which the mark is replicated as many times as needed prior to its insertion. All these issues are related to the retrieval capacity which is defined in the retrieval function.

The capacity is mostly specified in bits per second (bits/s). If another specifications are needed, like bits per frame or bits per kByte audio signal, it is simply converted. The embedding capacity can be divided into two classes, payload and embedded bits.

The data payload refers to the number of embedded bits of \( w \) into the audio signal, which are transmitted. It includes all bits, which are mostly more than the message \( m \) itself (like additional synchronization, protocol specific bits and/or the result of the coding function \( \text{cod} \)).

The embedding capacity refers to the embedded bits of the message \( m \) which is embedded into the audio signal. It is also specified in bits per second and can be converted into bits per frame or bits per kByte.

In the following, the notation based on \( m \) refers to the embedding capacity and based on \( w \) to the data payload.

**Embedding Transparency:** Transparency (or Imperceptibility) functions. Given a reference object \( S_{\text{ref}} \) and a test object \( S_{\text{test}} \) the transparency function \( T \) provides a measure of the perceptible distortion between \( S_{\text{ref}} \) and \( S_{\text{test}} \). Without loss of generality, such a function may take values in the closed interval \([0, 1]\) where 0 provides the worst case (the signals \( S_{\text{ref}} \) and \( S_{\text{test}} \) are so different that \( S_{\text{test}} \) cannot be recognized as a version
of $S_{\text{ref}}$ and 1 is the best case (an observer does not perceive any significant difference between $S_{\text{ref}}$ and $S_{\text{test}}$):

$$T(S_{\text{ref}}, S_{\text{test}}) \to [0, 1].$$

(3.6)

The relative transparency for a watermarking scheme $\Omega^*$ and a particular object $S$ is defined as:

$$T(S, S_{E}) \to \text{tra}_{E_{\text{rel}}}^{E}(\Omega^*, S)$$

(3.7)

where $S_{E}$ is obtained as per the embedding function Equation (3.4).

However, this definition of transparency is related to a particular object $S$. It is usually better to provide some absolute value of transparency which is not related to a particular object $S$. A definition of “absolute” transparency is related to a family $S$ of objects to be marked, which applies any of the following definitions:

- **Average transparency**:
  \[
  \text{tra}_{E_{\text{ave}}}^E(\Omega^*) = \frac{1}{|S|} \sum_{S \in S} \text{tra}_{E_{\text{rel}}}^E(\Omega^*, S).
  \]
  (3.8)

- **Maximum transparency**:
  \[
  \text{tra}_{E_{\text{max}}}^E(\Omega^*) = \max_{S \in S} \{\text{tra}_{E_{\text{rel}}}^E(\Omega^*, S)\}.
  \]
  (3.9)

- **Minimum transparency**:
  \[
  \text{tra}_{E_{\text{min}}}^E(\Omega^*) = \min_{S \in S} \{\text{tra}_{E_{\text{rel}}}^E(\Omega^*, S)\}.
  \]
  (3.10)

**Embedding Complexity**: Given a function $F$, the complexity of it can be measured. Thereby the effort or investment needed to embed the watermark is defined with embedding complexity. A measuring function $C$ is defined as $C(F)$ to measure the complexity of $F$. If it is adapted to the embedding function of $\Omega$, an audio signal is needed to measure the complexity and $F$ is the embedding Function $E$ from $\Omega$, $C(E, S)$. Depending on $C$, for example the computation cost of time, needed memory or IO operations, lines of code, etc. could be measured. The relative embedding complexity of a watermarking scheme $\Omega^*$ and a particular object $S$ is defined as:

$$C(E, S) \to \text{com}_{E_{\text{rel}}}^S(\Omega^*, S)$$

(3.11)

where $C$ is the complexity measure function. However, this definition of complexity depends on the audio signal $S$. Thereby, a normalization is needed to provide results independent on $S$. The normalization can be done with the audio signal and it length or with the embedded capacity. If the length of the audio signal is used for normalization, then the length can be time, data rate needed for streaming or file size on the storage. Which exactly is defined with the function size($S$). The normalization done by the embedding capacity measures the needed effort to embed one single bit. Note, that this normalization is only useable for n-bit watermarking schemes. In the following both normalizations are formalized.

$$\text{com}_{E_{\text{rel}}}^S(\Omega^*, S) = \frac{\text{com}_{E_{\text{rel}}}^E(\Omega^*, S)}{\text{size}(S)} = \frac{C(E, S)}{\text{size}(S)}$$

(3.12)
Note, that in this case a linear complexity depending on the length of \( S \) is assumed. If it is non-linear, then this function cannot be used to measure the complexity. Then, the normalization depending on the embedding capacity, introduced in the following can be used.

\[
\text{com}_{\text{rel}}^C(\Omega^*, S) = \frac{\text{com}_{\text{rel}}^c(\Omega^*, S)}{\text{cap}_E^c} = \frac{C(E, S)}{\text{cap}_E^c} \tag{3.13}
\]

Both definitions of complexity are related to a particular object \( S \). Similar to the embedding transparency, a definition of absolute values applies any of the following definitions:

- **Average complexity based on signal and capacity normalization:**

  \[
  \text{com}_{\text{ave}}^S(\Omega^*) = \frac{1}{|S|} \sum_{S \in S} \text{com}_{\text{rel}}^S(\Omega^*, S) \tag{3.14}
  \]

  \[
  \text{com}_{\text{ave}}^C(\Omega^*) = \frac{1}{|S|} \sum_{S \in S} \text{com}_{\text{rel}}^C(\Omega^*, S) \tag{3.15}
  \]

- **Maximum complexity for audio signal and capacity normalization:**

  \[
  \text{com}_{\text{max}}^S(\Omega^*) = \max_{S \in S} \{ \text{com}_{\text{rel}}^S(\Omega^*, S) \} \tag{3.16}
  \]

  \[
  \text{com}_{\text{max}}^C(\Omega^*) = \max_{S \in S} \{ \text{com}_{\text{rel}}^C(\Omega^*, S) \} \tag{3.17}
  \]

- **Minimum complexity for audio signal and capacity normalization:**

  \[
  \text{com}_{\text{min}}^S(\Omega^*) = \min_{S \in S} \{ \text{com}_{\text{rel}}^S(\Omega^*, S) \} \tag{3.18}
  \]

  \[
  \text{com}_{\text{min}}^C(\Omega^*) = \min_{S \in S} \{ \text{com}_{\text{rel}}^C(\Omega^*, S) \} \tag{3.19}
  \]

It is also possible to describe the average, maximum and minimum embedding complexity based on \( \text{com}_{\text{rel}}^c \). Therefore, the average embedding complexity for a given audio test set without normalization could be measured with \( \text{com}_{\text{ave}}^c \). Furthermore, the cover signal \( S \), whereby the highest \( \text{com}_{\text{max}}^c \) or lowest \( \text{com}_{\text{min}}^c \) embedding complexity were measured can be identified.

### 3.1.1.4 Detection and Retrieval Function

In the following, the focus is set on the question related to watermark or message detection and retrieval.

#### Detection Function

Given a test object \( S_{EA} \) (which is suspected to be a possibly attacked or modified version of the marked object \( S_E \)), a vector of embedding parameters \( p_E \), a vector \( p_D \in \mathcal{P}_D \) of detection parameters, the domain \( \mathcal{P}_D \) of all possible values of the detection parameters and, possibly,
the cover object $S$ and/or the embedded message $m$, a detection function $D$ can be defined in the following manner:

$$D(S_{EA}, p_E, p_D, [S, m]) \rightarrow \{0, 1\},$$

where $D$ returns 1 if $m$ is detected in $S_{EA}$ and 0 otherwise. Note that such a function can be used in either zero-bit or non-zero-bit watermarking schemes. Of course, in zero-bit watermarking schemes, the message $m$ is not used. Furthermore, if the watermarking scheme requires a public or private key for the detection process, then the key $k$ belonging to a key space $K (k \in K)$ is a component of the vector $p_E$, which is a parameter vector introduced in Equation (3.20).

### Retrieval Function

The definition of a retrieval function is only appropriate in non-zero-bit watermarking schemes. Given a test object $S_{EA}$ (which suspected to be a possibly attacked or modified version of the marked object), a vector of embedding parameters $p_E$, a vector $p_R \in P_R$ of retrieval parameters, the domain $P_R$ of all possible values of the retrieval parameters and, possibly, the cover object $S$ and/or the original message $m$, a retrieval function $R$ can be defined in the following manner:

$$m' = R(S_{EA}, p_E, p_R, [S_{EA}, m]),$$

where $m' \in M$ is an estimate of the embedded message referred to as the “identified message”.

In case of repeat coding, the message $m$ might have been embedded several times within the marked object. In this situation, some retrieval functions return all the different repetitions of the embedded message, whereas others use voting schemes and return just a single copy of the identified message. In the former case, the retrieved or identified message $m'$ may consist of a longer bit stream compared to the inserted message $m$. As part of $p_R$, the maximum number of multiple embedded $m$ is known and denoted as $p_{\text{max}}$. Furthermore, if the watermarking scheme requires a public or private key for the retrieval process, then the key $k$ belonging to a key space $K (k \in K)$ is a component of the vector $p_E$, which is a parameter vector introduced in Equation (3.21).

Note, also, that a detection function can be easily constructed from a retrieval function (but not conversely). Because of this, many multiple-bit watermarking schemes define retrieval functions instead of detection ones. Therefore, the following Table 3.1 introduces the dependencies between the retrieval and detection function and the zero-bit and $n$-bit watermark by introducing the watermark $w$ and message $m$.

<table>
<thead>
<tr>
<th>Watermark Type</th>
<th>Detection</th>
<th>Retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero-bit watermarking</td>
<td>$w$ in $S_{EA}$? (yes/no)</td>
<td>not available</td>
</tr>
<tr>
<td>$n$-bit watermarking</td>
<td>$w$ in $S_{EA}$? (yes/no)</td>
<td>$m'$</td>
</tr>
</tbody>
</table>

Table 3.1: Verification Cases
Classification according to the information needed by the detection or retrieval function

The watermarking schemes, which require the cover object $S$ in the detection function, are referred to as informed or non-blind. Some schemes require the original message $m$ and/or $p_E$ for detection or retrieval. These schemes are referred to as semi-blind. Finally, the schemes which do not require the original cover object $S$ nor the original message $m$ are referred to as blind watermarking scheme.

Retrieval Capacity The definition of retrieval capacity defines the capacity with respect to the retrieved message $m'$. First of all, zero-bit watermarking schemes do not transmit any message, since the watermark $w$ is just detected but a message $m$ is not retrieved. In such a case, the retrieval capacity of these schemes is zero.

For non-zero-bit watermarking schemes the retrieval capacity is considered after data extraction. Based on the retrieval function of Equation (3.21), the following retrieval capacity function is defined:

$$\text{cap}^*_R(\Omega^*, S_{EA}) = |m| - \sum_{i=1}^{\lfloor m \rfloor} m_i \oplus m'_i,$$

where $m = m_1 m_2 \ldots m_{\lfloor m \rfloor}$, $m' = m'_1 m'_2 \ldots m'_{\lfloor m \rfloor}$ and $\oplus$ depicts the exclusive or operation. This equation counts the number of correctly transmitted bits (those which are equal on both sides of the communication channel) and it is assumed that $m$ and $m'$ have exactly the same length (otherwise $m$ or $m'$ should be padded or cut in some manner).

In case of repeat coding, the retrieved message is several times longer than the embedded message: $m' = m'_1 m'_2 \ldots m'_{\lfloor m \rfloor} m'_1 m'_2 \ldots m'_{\lfloor m \rfloor} \ldots \ldots m'_{p_{\text{max}} \lfloor m \rfloor}$. In such a situation, the retrieval capacity should consider all the repetitions as follows:\footnote{It is not required that the number of message repetitions is an integer. The last repetition could be trimmed in the last few bits. For simplicity, the notation considers an integer number of repetitions.}

$$\text{cap}^*_R(\Omega^*, S_{EA}) = \sum_{j=1}^{p_{\text{max}}} \left[ |m| - \sum_{i=1}^{\lfloor m \rfloor} m_i \oplus m'_{ji} \right],$$

where $p_{\text{max}}$ is the counted number of maximal retrieved $m'$. In the sequel, no repeat coding is assumed for notational simplicity, but all the formulae can be easily extended to that case. If the watermark is not embedded multiple times, then $p_{\text{max}} = 1$, which is similar to Equation (3.22).

There are two relevant comments about this definition of relative capacity. The first is that usually this kind of measure is given in terms of the size of the cover object $S$:

$$\text{cap}_R(\Omega^*, S_{EA}) = \frac{\text{cap}^*_R(\Omega^*, S_{EA})}{\text{size}(S_{EA})},$$

and it is assumed that the sizes of $S$, and $S_{EA}$ are, at least, similar. This second definition provides measures such as bits per second or in bits of transmitted information.
per bit of the marked object. If the latter is used, a value in the interval \([0, 1]\) is obtained, where 1 means that all the transmitted bits are used for the message, which is the best case as capacity is concerned. The second comment is that \(\text{cap}_{\text{Rrel}}\) is relative to a given pair \(S_{EA}\) and \(S\). An absolute measure is provided below.

Another capacity measure can be defined in terms of the ratio of correctly recovered bits normalized by \(p_{\text{max}}\). If \(p_{\text{max}}\) is unknown, the measure of \(\text{cap}^\dagger_{\text{Rrel}}\) can also be used, but would result in highest, not normalized values:

\[
\text{cap}^\dagger_{\text{Rrel}}(\Omega^*, S_{EA}) = \frac{\text{cap}_{\text{Rrel}}^*(\Omega^*, S_{EA})}{|m|p_{\text{max}}}.
\]

### Detection/Retrieval Complexity

Refers to the effort needed to detect the watermark in given object \(S_{EA}\). Thereby, the introduced definition is similar to Equation (3.11) whereby the function \(F\) is replaced by the detection function \(D\) of \(\Omega\) for detection complexity and \(R\) for retrieval complexity.

\[
\text{com}^*_{\text{Drel}}(\Omega^*, S) = C(D, S_{EA}),
\]

\[
\text{com}^*_{\text{Rrel}}(\Omega^*, S) = C(R, S_{EA}),
\]

It is noted, that the complexity measured with the function \(C\) can be for example the computation cost of time, needed memory or IO operations. Similar to the embedding complexity, \(\text{com}^*_{\text{Drel}}\) and \(\text{com}^*_{\text{Rrel}}\) depend on a particular audio signal and its characteristic. Therefore, it can be normalized by the length of \(S_{EA}\) or by the length of retrieved message \(m'\). The last normalization is only for \(n\)-bit watermarking schemes usable.

\[
\text{com}^S_{\text{Drel}}(\Omega^*, S_{EA}) = \frac{\text{com}^*_{\text{Drel}}(\Omega^*, S_{EA})}{\text{size}(S_{EA})} = \frac{C(D, S_{EA})}{\text{size}(S)}
\]

\[
\text{com}^S_{\text{Rrel}}(\Omega^*, S_{EA}) = \frac{\text{com}^*_{\text{Rrel}}(\Omega^*, S_{EA})}{\text{size}(S_{EA})} = \frac{C(R, S_{EA})}{\text{size}(S)}
\]

Note, that in this case a linear complexity depending on the length of \(S_{EA}\) is assumed. If it is non-linear, then these functions cannot be used to measure the complexity. Then, the normalization depending on the retrieved capacity, only useable for \(n\)-bit watermarking schemes, is introduced in the following can be used.

\[
\text{com}^C_{\text{Rrel}}(\Omega^*, S_{EA}) = \frac{\text{com}^*_{\text{Rrel}}(\Omega^*, S_{EA})}{\text{size}(S_{EA})} = \frac{C(R, S_{EA})}{\text{cap}_{\text{Rrel}}}
\]

These complexity measures are related to a particular object \(S_{EA}\). Similar to the embedding complexity, definitions of absolute values (average, maximum and minimum) are introduced in the following.

- Average complexity based on signal and capacity normalization:

\[
\text{com}^S_{\text{Dave}}(\Omega^*) = \frac{1}{|S|} \sum_{S \in S} \text{com}^S_{\text{Drel}}(\Omega^*, S_{EA})
\]

\[
\text{com}^S_{\text{Rave}}(\Omega^*) = \frac{1}{|S|} \sum_{S \in S} \text{com}^S_{\text{Rrel}}(\Omega^*, S_{EA})
\]

\[
\text{com}^C_{\text{Rave}}(\Omega^*) = \frac{1}{|S|} \sum_{S \in S} \text{com}^C_{\text{Rrel}}(\Omega^*, S_{EA})
\]
• Maximum complexity for audio signal and capacity normalization:

\[
\begin{align*}
\text{com}_D^{\max}(\Omega^*) &= \max_{S \in S} \{ \text{com}_{D_{\text{rel}}}^S(\Omega^*, S_{\text{EA}}) \} \\
\text{com}_R^{\max}(\Omega^*) &= \max_{S \in S} \{ \text{com}_{R_{\text{rel}}}^S(\Omega^*, S_{\text{EA}}) \} \\
\text{com}_C^{\max}(\Omega^*) &= \max_{S \in S} \{ \text{com}_{C_{\text{rel}}}^S(\Omega^*, S_{\text{EA}}) \}
\end{align*}
\] (3.34) (3.35) (3.36)

• Minimum complexity for audio signal and capacity normalization:

\[
\begin{align*}
\text{com}_D^{\min}(\Omega^*) &= \min_{S \in S} \{ \text{com}_{D_{\text{rel}}}^S(\Omega^*, S_{\text{EA}}) \} \\
\text{com}_R^{\min}(\Omega^*) &= \min_{S \in S} \{ \text{com}_{R_{\text{rel}}}^S(\Omega^*, S_{\text{EA}}) \} \\
\text{com}_C^{\min}(\Omega^*) &= \min_{S \in S} \{ \text{com}_{C_{\text{rel}}}^S(\Omega^*, S_{\text{EA}}) \}
\end{align*}
\] (3.37) (3.38) (3.39)

Detection Success

To measure the overall success of a detection or retrieval function, the detection success function is introduced (see Equation (3.20)). Therefore, the connection to zero-bit an n-bit watermarking scheme are introduced as follows.

For zero-bit watermarking schemes, \(\text{det}_D(\Omega^*, S_{\text{EA}})\) returns 0, if the watermark could not be successful detected and 1 if the detection function was able to detect the watermark, see the following equation:

\[
\text{det}_D(\Omega^*, S_{\text{EA}}) = \begin{cases} 
0, & \text{no successful detection (negative),} \\
1, & \text{positive successful detection (positive).}
\end{cases}
\] (3.40)

To measure the successfully embedding rate over a test set \(S\), the average of \(\text{det}_D\) can be computed as follows:

\[
\text{det}_{D_{\text{ave}}}(\Omega^*) = \frac{1}{|S|} \sum_{S \in S} \text{det}_D
\] (3.41)

For n-bit watermarking schemes, it is important to know, if the watermark was successfully detected at least once (in case of multiple embedding). If, for example, a watermark scheme embeds the message \(m\) multiple times \((p_{\text{max}})\), and the retrieval function \(\text{cap}_{R_{\text{rel}}}^\star\) returns, that 10% are positive retrievable, then it is unknown, which \(m_i\) are affected. Therefore, it is useful to define a successful detection, if at least one embedded message could be retrieved positively, which is introduced in the following equation.

\[
\text{det}_R(\Omega^*, S_{\text{EA}}) = \begin{cases} 
1, & \exists j \in \{1, \ldots, p_{\text{max}}\} : \sum_{i=1}^{\left| m_i \right|} m_{ij}^r \oplus m_{ij} = 0, \\
0, & \text{otherwise.}
\end{cases}
\] (3.42)

Note that this is not the only possible definition of the detection function in case of repeat coding. For example, another definition could be the following:

\[
\text{det}_{R_{\tau}}(\Omega^*, S_{\text{EA}}) = \begin{cases} 
1, & \text{if } \text{cap}_{R_{\text{rel}}}^\tau(\Omega^*, \tilde{S}) \geq \tau, \\
0, & \text{otherwise.}
\end{cases}
\] (3.43)
i.e. detection is reported if the ratio of correctly recovered bits is above some threshold $\tau$ (which is equal to or close to 1).

To measure the successfully embedding rate over a test set, the average of $\text{det}_R$ and $\text{det}_{R\tau}$ can be computed as follows:

$$\text{det}_{R\text{ave}}(\Omega^*) = \frac{1}{|S|} \sum_{S \in \mathcal{S}} \text{det}_R$$  \hspace{2cm} (3.44)

$$\text{det}_{R\tau\text{ave}}(\Omega^*) = \frac{1}{|S|} \sum_{S \in \mathcal{S}} \text{det}_{R\tau}$$  \hspace{2cm} (3.45)

### 3.1.1.5 Watermark Attacking Functions

An attacking function or attack $A$ distorts a marked object $S_E$ and providing a modified version $S_{EA}$ (test object) aiming to destroy or weaken the embedded information. $S_{EA}$ is often referred to as the attacked object:

$$S_{EA} = A(S_E, p_A),$$  \hspace{2cm} (3.46)

where $p_A \in \mathcal{P}_A$ is a set of attacking parameters and the function $A$ a selected attack $A \in \mathcal{A}$.

Usually, a family of attacking functions $A_{i,j} \in \mathcal{A}$ exist which may be applied to some object, where $i$ identifies an attack and $j$ is a related parameter combination $(p_{A_{i,j}})$. It is assumed that attacks are “simple”. If composite (or concatenation) attacks $A_{i,1} \circ A_{i,2} \circ \cdots \circ A_{1,j} \circ A_{2,1} \circ \cdots \circ A_{2,j} \circ \cdots \circ A_{i_{\text{max}},j_{\text{max}}}(S_{EA})$ are possible, these should be incorporated explicitly into the attack family $\mathcal{A}$. Note that different attack domains $\mathcal{A}$ can be defined according to different scenarios. The concatenation of such single attacks is often referred as profile attacks [LD04, LDSV05, LDLD05, LD06a, WET] and discussed later.

**Robustness/Fragility** A watermarking scheme $\Omega$ is defined to be robust if the detection function $D$ for zero-bit watermarking schemes or if the retrieval function $R$ for $n$-bit watermarking schemes is able to recover the mark even when the attacks contained in a family $\mathcal{A}$ are applied to the marked object. In contrast, $\Omega$ is defined to be fragile if the detection function $D$ or retrieval function $R$ is not able to recover the mark after applying an attack $A$. Furthermore, the fragility measure divides into the type of attackers because depending on the type of attacker, the watermark should be fragile. If for example a “lossy compression” is performed from a non-malicious attacker, the watermark should be alive after this process. In contrast, if a malicious attacker uses “lossy compression” as attack against the watermark, then the fragile watermark should be broken. To distinct between these fragilities, different levels of fragility exist, which provide a classification briefly introduced in the following description:

**Bit-fragile:** The watermark should be are broken, if at least one single bit in the whole marked audio signal is switched. It does not survive any signal processing operations and can be used to prove the bit integrity of the content.

**Content-fragile:** The watermark is more robust than bit-fragile watermarks. They should survive signal processing operations like lossy compression or other ones,
where the content itself is not manipulated. A content-fragile watermark is destroyed, if the content is manipulated like cutting of content or inserting content into the audio signal.

It is noted, that the level of fragility depends on the application field used later. Furthermore, the application scenario decides, if the watermark should be robust to survive an attacks and to be fragile to detect manipulations occurred by an attack.

The definition of robustness only classifies watermarking schemes in two categories: robust or not robust (fragile) and does not limit the distortion introduced in the marked object by the attacking functions. For example, the attacking function $A_{i,j}(\hat{S}) = \emptyset$, where $\emptyset$ means that the object is deleted, always erases the mark since it deletes the signal itself. However, the attack might certainly produce very bad transparency results: $T(\tilde{S}, \emptyset) \approx 0$. Thus, although the attack is successful in terms of erasing an embedded mark, it would be considered useless for most typical watermarking applications as the overall object quality decreases. If an attack exists, which destroys the embedded mark and, at the same time, produces little distortion, this means that the watermarking scheme is not robust enough and should be enhanced. For this reason, we establish a relationship between robustness and attacking transparency by means of a quantitative robustness measure, in the following definition.

**Robustness Measure**

The robustness measure $\text{rob}_{rel}$ of a watermarking scheme is a value in the closed interval $[0, 1]$, where 0 is the worst possible value (the scheme is not robust for the signal $S$) and 1 is the best possible value (the method is robust for the signal $S$). There is a difference, depending on whether the bit error rate (BER) or byte error rate (BYR) is used to measure the robustness. If the robustness is measured based on the byte error rate $\text{rob}_{byte}$, then a given watermarking scheme is classified as robust if the bytes of the embedded message (characters) are correctly retrieved. This measurement is similar to the Levenstein distance [Lev66], which works and measured a distance between two given strings. It is useful in applications scenarios that need to determine how similar two strings are. Another robustness measure function based on the bit error rate $\text{rob}_{bit}$ returns the percentage robustness of the watermarking scheme measured over the whole attacking and test set and is based on the bit changes within the retrieved message. This measurement is similar to the Hamming distance [Ham50] based on bit-strings. Hence, a watermarking scheme is classified as not robust, if more than $\nu$ numbers of retrieved bits are destroyed and the transparency of the attacks if higher than $\tau$. For zero-bit watermarking schemes no retrieval function exists and no classification based on bit or byte error rates are possible. To simplify matters, the robustness measure for zero-bit watermarking schemes is always classified to $\text{rob}_{byte}$.

The following example motivates the distinction between the robustness measure based on bit and byte error rate. If the message $m=“123”$, with 3 bytes and $3^8=24$ bits, is embedded and after attacking, the last 6 bits are destroyed and incorrectly retrieved, then the byte error rate returns, that 2 bytes are correct (the first two) and one is false (the last), which has a value of $\frac{2}{3} = 0.67$. The bit error rate returns, that 18 bits are correct (the first) and 6 bits are false (the last), which has a value of $\frac{6}{24} = 0.25$. If now
the 1., 2., 8., 9., 16. and 17. bit are destroyed, then the byte error rate returns, that all bytes (characters) are false and the result has a value of \( \frac{3}{3} = 1.0 \) and this shows, that 100% of the bytes are destroyed. In contrast, the bit error rate returns, that 18 bits are correct retrieved and 6 bits are wrong, which has a value of \( \frac{6}{24} = 0.25 \). Although the bit error rate does not change to the first example, differences are apparent in the byte error rate. Therefore, the following equations introduce the robustness for n-bit watermarking schemes divided into \( \text{rob}_{\text{byte}} \) and \( \text{rob}_{\text{bit}} \) and for zero-bit watermarking schemes only for \( \text{rob}_{\text{byte}} \). The two robustness measures \( \text{rob}_{\text{byte}} \) and \( \text{rob}_{\text{bit}} \) returns completely different robustness values. It is introduced to show, that different approaches are possible and depending on test goals, choices are to be made to select the measure function. It is noted, that different measure methods are available to measure the robustness, i.e. based on \( \text{det}_{R} \) in relation to attacking transparency.

The following function relates robustness based on the byte error rate to transparency for a zero-bit and n-bit watermarking scheme as follows, given \( S_{EA} = A_{i,j}(S_{E}) \):

\[
\text{rob}_{\text{rel}} \left( \Omega^{*}, S_{E} \right) = 1 - \max_{A_{i,j} \in A} \left\{ T \left( S_{E}, S_{EA} \right) : \text{det}_{D} \left( S_{EA}, p_{E}^{\text{opt}}, p_{D}^{\text{opt}}, p_{\text{cod}}, [S, m] \right) = 0 \right\},
\]

(3.47)

and for a n-bit watermarking scheme:

\[
\text{rob}_{\text{rel}} \left( \Omega^{*}, S_{E} \right) = 1 - \max_{A_{i,j} \in A} \left\{ T \left( S_{E}, S_{EA} \right) : \text{det}_{R} \left( S_{EA}, p_{E}^{\text{opt}}, p_{D}^{\text{opt}}, p_{\text{cod}}, [S, m] \right) = 0 \right\},
\]

(3.48)

And the robustness based on the bit error rate related to the transparency for n-bit watermarking schemes is given as:

\[
\text{rob}_{\text{ave}} \left( \Omega^{*} \right) = \frac{1}{|S_{EA}| |A|} \sum_{S \in S} \sum_{A_{i,j} \in A} \left\{ 1, \left( \text{cap}_{R_{\text{rel}}} < \tau \right) \land (\text{tra}_{A_{\text{rel}}} > \nu) \right\},
\]

(3.49)

That is, given a marked object \( S_{E} \) and all the attacks which attack the watermark, even for optimal embedding and detection parameters \( (p_{E}^{\text{opt}}, p_{D}^{\text{opt}}) \), the one which produces less distortion in the marked object \( S_{E} \) determines how robust the scheme is. If none of the attacks in the family \( A \) erases the embedded mark, then this measure is (by definition) equal to 1 (the best possible value).

The functions provided in Equation (3.47), Equation (3.48) and Equation (3.49) measure robustness in a worst case sense. When the security of a system is to be assessed, it is usually considered that a given system is as weak as the weakest of its components. Similarly, Equation (3.48) establishes that the worst possible attack (in the sense that the mark is erased but the attacked signal preserves good quality) in a given family determines how robust the watermarking scheme \( \Omega \) is. If the best (maximum) transparency amongst all the attacks which destroy the mark is 0.23, then the robustness of the method as given by Equation (3.48) is \( 1 - 0.23 = 0.77 \).

However, the functions of Equation (3.47) and Equation (3.48) are relative to a given object \( S_{EA} \) (hence the use of the subindex "rel") but usually to define the robustness of a watermarking scheme as an inherent property not related to any particular object, but to a family or collection of objects. This may be referred to as the absolute
robustness \(\text{rob}_\text{byte}^{\text{rel}}\) which can be defined in several ways. Given a family \(\mathcal{S}\) of cover objects, and their corresponding marked objects \(\mathcal{S}_E\) obtained by means of the embedding Equation (3.4), the absolute robustness based on bit and byte error rate can be defined according to different criteria, for example:

- **Average robustness based on byte error rate:**
  \[
  \text{rob}^{\text{byte}}_{\text{ave}}(\Omega^*, \mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \text{rob}^{\text{byte}}_{\text{rel}}(\Omega^*, S_E).
  \]

- **Minimum robustness (worst case approach) based on byte error rate:**
  \[
  \text{rob}^{\text{byte}}_{\text{min}}(\Omega^*) = \min_{S \in \mathcal{S}} \text{rob}^{\text{byte}}_{\text{rel}}(\Omega^*, S_E).
  \]

- **Probabilistic approach based on byte error rate:**
  \[
  \text{rob}^{\text{byte}}_{\text{prob}}(\Omega^*, r) = 1 - \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \text{rob}^{\text{byte}}_{\text{rel}}(\Omega^*, S_E < r).
  \]

where \(p\) stands for “probability” and \(r\) is some given threshold. For example, if \(r = 0.75\) and \(\text{rob}_{\text{prob}} = 0.9\), this means that 90% of the objects in \(\mathcal{S}\) provide a relative robustness greater than or equal to 0.75 for the scheme \(\Omega\).

Although a maximum robustness measure could thus be defined, it does not seem to have any applicability, since worst or average cases are often reported as robustness is concerned.

**Attacking Transparency** The definition of a relative transparency for the attacking process for a watermarking scheme \(\Omega^*\) and a particular object \(S\) is as. Two different measures can be provided. The first is the transparency of the attacked object \((S_{E_A})\) with respect to the marked object \((S_E)\), which is the most obvious one:

\[
T(S_E, S_{E_A}) \rightarrow \text{tra}_{A_{rel}}(\Omega^*, S_E, S_{E_A}),
\]

where \(S_E\) is obtained as per the embedding function Equation (3.4) and \(S_{E_A} = A_{i,j}(S_E), p_{A_{i,j}}\) for some attack.

A second measure could be provided to define the transparency of the attacked signal with respect to the original signal and based \(p_{A_{i,j}}\) parameter:

\[
T(S, S_{E_A}) \rightarrow \text{tra}^*_A_{rel}(\Omega^*, S, S_{E_A}).
\]

The usefulness of this measure might not be obvious, but it must be taken into account that a given attack could result in an attacked signal which is closer to the original object \(S\) than to the marked object \(S_E\). In such a case, the attack could provide an object which is even better than the marked one as far as transparency is concerned and the mark could be erased. Hence, this measure should also be considered in some situations.

It is usually better to provide some absolute value of transparency which is not related to a particular object \(S\). Therefore, any of the following definitions can be applied:
• Average transparency:

\[
\text{tra}_{A_{\text{ave}}}^{*}(\Omega^{*}) = \frac{1}{|S| |A|} \sum_{S \in S} \sum_{A_{i,j} \in A} \text{tra}_{\text{rel}}(\Omega^{*}, S, S_{EA}).
\]

(3.55)

\[
\text{tra}_{A_{\text{ave}}}^{*}(\Omega^{*}) = \frac{1}{|S| |A|} \sum_{S \in S} \sum_{A_{i,j} \in A} \text{tra}_{\text{rel}}(\Omega^{*}, S, S_{EA}).
\]

(3.56)

• Maximum transparency:

\[
\text{tra}_{A_{\text{max}}}^{*}(\Omega^{*}) = \max_{S \in S} \left\{ \max_{A_{i,j} \in A} \{ \text{tra}_{\text{rel}}(\Omega^{*}, S, S_{EA}) \} \right\}.
\]

(3.57)

\[
\text{tra}_{A_{\text{max}}}^{*}(\Omega^{*}) = \max_{S \in S} \left\{ \max_{A_{i,j} \in A} \{ \text{tra}_{\text{rel}}(\Omega^{*}, S, S_{EA}) \} \right\}.
\]

(3.58)

• Minimum transparency:

\[
\text{tra}_{A_{\text{min}}}^{*}(\Omega^{*}) = \min_{S \in S} \left\{ \min_{A_{i,j} \in A} \{ \text{tra}_{\text{rel}}(\Omega^{*}, S, S_{EA}) \} \right\}.
\]

(3.59)

\[
\text{tra}_{A_{\text{min}}}^{*}(\Omega^{*}) = \min_{S \in S} \left\{ \min_{A_{i,j} \in A} \{ \text{tra}_{\text{rel}}(\Omega^{*}, S, S_{EA}) \} \right\}.
\]

(3.60)

**Attacking Capacity** The relative attacking capacity \(\text{cap}_{\text{rel}}^{*}\) of a watermarking scheme and object is similar to the retrieval capacity \(\text{cap}_{\text{rel}}\) but after applying an attack \(A_{i,j}\). If the attack does not change the audio signal (like the “nothing” attack), then the result of \(\text{cap}_{\text{rel}}^{*}\) is equal to the result of \(\text{cap}_{\text{rel}}\). Furthermore, \(\text{cap}_{\text{rel}}^{*}\) based on \(\text{cap}_{\text{rel}}^{*}\) can be computed. Both are defined as:

\[
\text{cap}_{\text{rel}}(\Omega^{*}, S_{EA}) = \text{cap}_{\text{rel}}(\Omega, S_{EA}) : A_{i,j},
\]

(3.61)

\[
\text{cap}_{\text{rel}}^{*}(\Omega^{*}, S_{EA}) = \text{cap}_{\text{rel}}^{*}(\Omega, S_{EA}) : A_{i,j}.
\]

(3.62)

Both definitions are related to a particular object, therefore, they can be related to a family of attacks \(A\) and a family of objects \(S\) as follows:

• Average capacity:

\[
\text{cap}_{A_{\text{ave}}}^{*}(\Omega^{*}) = \frac{1}{|S| |A|} \sum_{S \in S} \sum_{A_{i,j} \in A} \text{cap}_{\text{rel}}^{*}(\Omega^{*}, S_{EA}),
\]

(3.63)

\[
\text{cap}_{A_{\text{ave}}}^{*}(\Omega^{*}) = \frac{1}{|S| |A|} \sum_{S \in S} \sum_{A_{i,j} \in A} \text{cap}_{\text{rel}}^{*}(\Omega^{*}, S_{EA}).
\]

(3.64)

• Maximum capacity:

\[
\text{cap}_{A_{\text{max}}}^{*}(\Omega^{*}) = \max_{S \in S} \left\{ \max_{A_{i,j} \in A} \{ \text{cap}_{\text{rel}}^{*}(\Omega^{*}, S_{EA}) \} \right\},
\]

(3.65)

\[
\text{cap}_{A_{\text{max}}}^{*}(\Omega^{*}) = \max_{S \in S} \left\{ \max_{A_{i,j} \in A} \{ \text{cap}_{\text{rel}}^{*}(\Omega^{*}, S_{EA}) \} \right\}.
\]

(3.66)
• Minimum capacity:

\[
\text{cap}_{A_{\min}}(\Omega^*) = \min_{S \in S} \left\{ \min_{A_{i,j} \in A} \{\text{cap}_{R_{rel}}(\Omega^*, S_E A)\} \right\}, \quad (3.67)
\]

\[
\text{cap}^*_A_{\min}(\Omega^*) = \min_{S \in S} \left\{ \min_{A_{i,j} \in A} \{\text{cap}^*_{R_{rel}}(\Omega^*, S_E A)\} \right\}. \quad (3.68)
\]

Therefore, based on the retrieved capacity \(\text{cap}_{R_{rel}}\) and \(\text{cap}^*_{R_{rel}}\) from \(R\), the attacking capacity is introduced as shown above. It is also possible to describe the attacking capacity based on the other defined retrieving capacity \(\text{cap}^i_{R_{rel}}\).

**Attacking Complexity**

The definition of attacking complexity is and a particular object is similar to the definition of embedding complexity Equation (3.11). A measuring function \(C\) is defined as \(C(F)\) to measure the complexity of a function \(F\). For attacking complexity, \(F\) is related to an attacking function \(A_{i,j}\) and an audio signal \(S_E\). Depending on \(C\), for example the computation cost of time, needed memory or IO operations could be measured. The relative attacking complexity of an attack \(A_{i,j}\) and a particular object \(S\) is defines as:

\[
C(A_{i,j}, S_E) \rightarrow \text{com}^*_{A_{rel}}(A_{i,j}, S_E), \quad (3.69)
\]

where \(C\) is the complexity measure function. However, this definition of complexity depends on the audio signal \(S_E\). Thereby, a normalization is needed to provide results independent on \(S_E\). Similar to the embedding complexity, a normalization is needed to provide comparability independent on the audio signal. Therefore, the following equation introduces the normalized attacking complexity:

\[
\text{com}_{A_{rel}}(A_{i,j}, S_E) = \frac{\text{com}^*_{A_{rel}}(A_{i,j}, S_E)}{\text{size}(S_E)} = \frac{C(A_{i,j}, S_E)}{\text{size}(S_E)}. \quad (3.70)
\]

Note, that in this case a linear complexity depending on the length of \(S_E\) is assumed. If it is non-linear, then this function cannot be used to measure the complexity. This definition of complexity is related to a particular object \(S_E\). Similar to the embedding complexity, a definition of absolute values applies any of the following definitions:

- **Average complexity based on signal normalization:**

\[
\text{com}_{A_{ave}}(A) = \frac{1}{|A|} \sum_{A \in A} \text{com}_{A_{rel}}(A_{i,j}, S_E) \quad (3.71)
\]

(3.72)

- **Maximum complexity for audio signal normalization:**

\[
\text{com}_{A_{max}}(A) = \max_{A \in A} \{\text{com}_{A_{rel}}(A_{i,j}, S_E)\} \quad (3.73)
\]

(3.74)

- **Minimum complexity for audio signal normalization:**

\[
\text{com}_{A_{min}}(A) = \min_{A \in A} \{\text{com}_{A_{rel}}(A_{i,j}, S_E)\} \quad (3.75)
\]

(3.76)
Relationship between capacity and robustness

Taking the definitions into account provided above, it may seem that capacity and robustness are not related, because the formulae provided do not involve both of them in a particular equation. However, it must be taken into account that robustness is related to the detection function $\det_D$ or retrieval function $\det_R$. Following that successful detection after attacking $\det_A$ for a specific attack or $\det_{A_{ave}}$ for an average value over a set of attacks with $p_A$ can be described for zero-bit watermarking schemes as:

$$\det_A = \frac{1}{|\mathcal{S}| |\mathcal{A}|} \sum_{S \in \mathcal{S}} \det_D, \text{ for a specific attack } A_{i,j} \quad (3.77)$$

and for n-bit watermarking schemes as:

$$\det_A = \frac{1}{|\mathcal{S}| |\mathcal{A}|} \sum_{S \in \mathcal{S}} \det_R, \text{ for a specific attack } A_{i,j} \quad (3.78)$$

The average detection success for zero-bit watermarking schemes is:

$$\det_{A_{ave}} = \frac{1}{|\mathcal{S}| |\mathcal{A}|} \sum_{S \in \mathcal{S}} \sum_{A_{i,j} \in \mathcal{A}} \det_D \quad (3.79)$$

and for n-bit watermarking schemes as:

$$\det_{A_{ave}} = \frac{1}{|\mathcal{S}| |\mathcal{A}|} \sum_{S \in \mathcal{S}} \sum_{A_{i,j} \in \mathcal{A}} \det_R \quad (3.80)$$

With $\det_{A_{ave}}$ the normalized successful detection after attacking can be measured and the result is in the range $[0, 1]$. Hence, based on the detection success $\det_{R_{\tau}}$ the detection success after attacking can be measured as shown above. The result would be $\det_{A_{\tau}}$ for a specific value or $\det_{A_{ave\tau}}$ as an average value over a given test set.

3.1.1.6 Other Properties – Invertibility, Verification, Security

For watermarking schemes, other properties exists. In the following description, these properties are introduced and formalized.

**Invertibility:** refers to the property of a watermarking scheme which has the possibility to remove the watermark $w$ from the marked audio signal $S_E$ completely to receive audio signal $S'$ and if $\Omega$ is invertible, then $S = S'$. To provide this feature, the watermarking algorithms must provide special embedding techniques. Furthermore, secret keys are mostly used to protect the original content from unauthorized access. The measured value of invertibility for a watermarking scheme $\Omega^*$ is a boolean value. If this value is 0, then $\Omega^*$ cannot remove $w$ from the marked object. If $\Omega$ can remove $w$ completely and $S = S'$, then 1 is returned.

$$\text{inv}(\Omega^*, S_E) = \begin{cases} 
0 & ((\Omega^*, S_E) \rightarrow S') \land (S \neq S') = \text{true} \\
1 & ((\Omega^*, S_E) \rightarrow S') \land (S = S') = \text{true} 
\end{cases} \quad (3.81)$$
Verification: described the type of the detection/retrieval function $D, R$ (see page 17) which require information. Therefore three classifications are available:

Non-blind: If the watermarking scheme requires the cover object $S$, then it is associated as non-blind watermarking scheme. Often, this type of watermark scheme is referred as informed watermarking scheme. Mostly, the watermark detector/retriever is only usable from a defined group of people, which hide the watermark detector and the required original signal $S$.

Informed: If the watermarking scheme requires the embedded message $m$, the embedding parameters $p_E$ or other additional information (except the original signal $S$) for detection or retrieval, then the watermarking scheme is associated to this group. Often, watermarking schemes where the embedding function creates a data file needed for detection/retrieval, are associated to this type of verification.

Blind: If the watermarking scheme does not require the original signal nor additional information (e.g. $m$ or $p_E$), then the watermarking scheme is associated to this group.

The verification (ver) is defined as list $\{0, 0.5, 1\}$, whereby the 1 is associated with non-blind, a 0.5 with informed and a 0 with blind. The formalization is introduced in the following equation.

$$\text{ver}(\Omega^*, S_E) = \begin{cases} 
0 & (\Omega^*, S_E) \text{ is non-blind} \\
0.5 & (\Omega^*, S_E) \text{ is informed} \\
1 & (\Omega^*, S_E) \text{ is blind}
\end{cases} \quad (3.82)$$

Security: described the security of the embedded watermark against specific security attacks. From the briefly introduced different definitions of the security of digital watermarking schemes, a formalization is required to measure on one hand the security and on the other hand to provide comparability. Thereby, it must be possible, to provide intra-algorithm evaluation and analysis by using one selected watermarking scheme $\Omega$ with different parameter settings and inter-algorithm evaluation and analysis to compare different watermarking schemes $\Omega_1, \ldots, \Omega_j$, whereby $j$ denotes the number of watermarking schemes, each other. Therefore, exemplary security formalizations to provide a measurement of the watermark security are selected. The selected security formalizations are examples and should motivate the general principle of the watermark security measurements. Thereby, we claim to measure the security of an embedded watermark in an interval $[0, 1]$, whereby 0 denotes the worst and 1 the best possible case.

In the following equation is the definition from Bas et al. [BcC06] used, to formalize a measurement of the relative subspace security of a watermarking scheme $\text{sec}_{\text{rel}}^{\text{subs}}$.

$$\text{sec}_{\text{rel}}^{\text{subs}}(\Omega^*, S) = \begin{cases} 
0 & (\Omega^*, S) \text{ is insecure} \\
0.5 & (\Omega^*, S) \text{ is key-secure} \\
1 & (\Omega^*, S) \text{ is subspace-secure}
\end{cases} \quad (3.83)$$

Thereby, $\Omega^*$ denotes an instance of the watermarking scheme with a specific, selected parameter set and $S$ is the original unmarked signal. If the security is measured over a huge audio test set $\mathcal{S}$ with $S \in \mathcal{S}$, then the average, minimum and maximum subspace security is measured as follows:
- Average subspace security:

\[
\text{sec}_{\text{ave}}^{\text{subs}}(\Omega^*) = \frac{1}{|S|} \sum_{S \in \mathcal{S}} \text{sec}_{\text{rel}}^{\text{subs}}(\Omega^*, S).
\]  

(3.84)

- Maximum subspace security:

\[
\text{sec}_{\text{max}}^{\text{subs}}(\Omega^*) = \max_{S \in \mathcal{S}} \left\{ \text{sec}_{\text{rel}}^{\text{subs}}(\Omega^*, S) \right\}.
\]  

(3.85)

- Minimum subspace security:

\[
\text{sec}_{\text{min}}^{\text{subs}}(\Omega^*) = \min_{S \in \mathcal{S}} \left\{ \text{sec}_{\text{rel}}^{\text{subs}}(\Omega^*, S) \right\}.
\]  

(3.86)

If the security evaluation should be done, by measure the collusion residence of \( \Omega^* \) with a given parameter set and the signal \( S \), then the following security evaluation should be performed. Thereby, the relative collusion resistance is measured. It is noted, that \( \Omega^* \) with given parameter set and \( S \) can be resistant or cannot be resistant against a collusion attack.

\[
\text{sec}_{\text{rel}}^{\text{coll}}(\Omega^*, S) = \begin{cases} 
0 & (\Omega^*, S) \text{ is not collusion secure} \\
1 & (\Omega^*, S) \text{ is collusion secure}
\end{cases}
\]  

(3.87)

If the security is measured over a huge audio test set \( \mathcal{S} \) with \( S \in \mathcal{S} \), then the average, minimum and maximum subspace security is measured as follows:

- Average collusion security:

\[
\text{sec}_{\text{ave}}^{\text{coll}}(\Omega^*) = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \text{sec}_{\text{rel}}^{\text{coll}}(\Omega^*, S).
\]  

(3.88)

- Maximum collusion security:

\[
\text{sec}_{\text{max}}^{\text{coll}}(\Omega^*) = \max_{S \in \mathcal{S}} \left\{ \text{sec}_{\text{rel}}^{\text{coll}}(\Omega^*, S) \right\}.
\]  

(3.89)

- Minimum collusion security:

\[
\text{sec}_{\text{min}}^{\text{coll}}(\Omega^*) = \min_{S \in \mathcal{S}} \left\{ \text{sec}_{\text{rel}}^{\text{coll}}(\Omega^*, S) \right\}.
\]  

(3.90)

The introduced concept to measure the subspace and collusion security of a watermarking scheme should motivate to adapt all other security definitions and to enhance the simple measure methods. After setting up the set of all security measurements \( \mathcal{L} \), with \( \text{sec}_{\text{subs}} \in \mathcal{L} \) and \( \text{sec}_{\text{coll}} \in \mathcal{L} \), the general, total security of a watermarking scheme \( \Omega^* \) can be measured by computing the relative total security \( \text{sec}_{\text{rel}}^{\text{tot}} \) of a given \( S \) as follows:

\[
\text{sec}_{\text{rel}}^{\text{tot}}(\Omega^*, S) = \frac{1}{|\mathcal{L}|} \sum_{\text{sec}_{\text{rel}} \in \mathcal{L}} \text{sec}_{\text{rel}}^{\text{tot}}(\Omega^*, S)
\]  

(3.91)

Whereby \( \text{sec}_{\text{rel}}^{\text{tot}} \) defines each relative security measurement provided by \( \mathcal{L} \), for example, \( \text{sec}_{\text{rel}}^{\text{subs}} \) or \( \text{sec}_{\text{rel}}^{\text{coll}} \) and all other security measurements defined in the security set \( \text{sec} \). If the average total security \( \text{sec}_{\text{ave}}^{\text{tot}} \), maximum \( \text{sec}_{\text{max}}^{\text{tot}} \) and minimum \( \text{sec}_{\text{min}}^{\text{tot}} \) are measured, then the following definition are used.
• Average total security:
\[
\text{sec}^{\text{ave}}_{\text{tot}}(\Omega^*) = \frac{1}{|\mathcal{S}| |\mathcal{L}|} \sum_{S \in \mathcal{S}} \sum_{\text{sec}^{\text{ave}} \in \mathcal{L}} \text{sec}^{\text{ave}}_*(\Omega^*, S)
\]  
(3.92)

• Maximum total security:
\[
\text{sec}^{\text{max}}_{\text{tot}}(\Omega^*) = \max_{S \in \mathcal{S}} \left\{ \max_{\text{sec}^{\text{rel}}_{\text{max}}} \{ \text{sec}^{\text{rel}}_*(\Omega^*, S) \} \right\}
\]  
(3.93)

• Minimum total security:
\[
\text{sec}^{\text{min}}_{\text{tot}}(\Omega^*) = \min_{S \in \mathcal{S}} \left\{ \min_{\text{sec}^{\text{rel}}_{\text{min}}} \{ \text{sec}^{\text{rel}}_*(\Omega^*, S) \} \right\}
\]  
(3.94)

### 3.1.1.7 Realization of Security Measurements

In this subsection, the practical realization of the exemplary selected subspace security measurement \(\text{sec}^{\text{subs}}\) is introduced. Therefore, different security attacks, introduced in the literature, are selected, briefly introduced and its impact on \(\text{sec}^{\text{subs}}\) described.

One typical, well known and time consumption security attack, which focuses on the secret key of the watermarking system, is the brute force key searching attack \((A_1)\). Thereby all possible keys \(k\) are used to detect the watermark and to retrieve the secret message. Another security attack \((A_2)\) focuses on asymmetrical key pairs \((k_{\text{priv}}\text{ and } k_{\text{pub}})\), whereby one is used for embedding and the other for retrieving the message. The attack tries to derive the private key \(k_{\text{priv}}\) from the known public key \(k_{\text{pub}}\). The third attack \(A_3\), so called sensitivity attack, which is introduced in [CL97] consists in the iterative modification of the coefficients of the watermarked vector to estimate the boundary of the detection region by observing the outputs of the detector. Thereby, it is assumed, that the knowledge of the detection region implies the knowledge of the watermark. Other security attacks, summarized in [CFF05] focus on the Diffie-Hellman terminology and provide three types of attacks. The first attack \(A_4\), called Watermark Only Attack (WOA), is performed, when the attacker has only access to one or more the watermarked signals. In contrast, the second attack \(A_5\), called Known Message Attack (KMA), is performed, when the attacker has access to the watermarked signal and the embedded message. This type of attack is used, when the attacker knows the embedded message like, for example, by the application field of copyright identification or annotation. The third type of attack \(A_6\), called Known Original Attack (KOA), is performed, when the attacker has access to the watermarked signal and the original (unmarked) signal. This type of attack can be used, if the watermarking scheme is classified as non-blind, where the detection is only possible with the original signal.

Let these few introduced security attacks taken to show the impact of them on the subspace security. Therefore, Table 3.2 summarizes the exemplary selected security attacks and shows the impact on the three levels of subspace security.
Table 3.2: Association of Subspace Security and selected Security Attacks

<table>
<thead>
<tr>
<th>secsubs</th>
<th>A_1</th>
<th>A_2</th>
<th>A_3</th>
<th>A_4</th>
<th>A_5</th>
<th>A_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subspace-secure</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Key-secure</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>○</td>
<td>∗</td>
<td>∗</td>
</tr>
<tr>
<td>Insecure</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>○</td>
<td>∗</td>
<td>∗</td>
</tr>
</tbody>
</table>

Thereby, a ✓ means, that the security feature is provided and cannot be successful attacked. A × means, that the specific attack works successful in this security class. A ○ denotes, that the information cannot be given and needs to be analyzed in detail.

The evaluation of digital audio watermarks is a manifold process. To evaluate the robustness of watermarking algorithms, there are single and/or profile attacks available [LDLD05, LD04]. Other properties like complexity or capacity are often neglected. Therefore, from the introduced parameters of watermarking schemes, a methodology is derived and introduced. Based on it, profiles and their parameters used to enhance the benchmarking are introduced.

Many processes of evaluating digital watermarking algorithms have been developed. The general goal of all available systems is to provide a comparability of the watermarking algorithms. The differences on the systems are the strategies – what and how they are doing the evaluation. Another fact is, that the procedure is not clearly defined to provide a good methodology for the general evaluation process.

The following section introduces the methodology of benchmarking. Therefore are the different benchmarking concepts classified and introduced in detail. Furthermore, the benchmarking concepts by using profiles are described. To introduce the benchmarking of watermarking algorithms, two different point of views are introduces. Firstly, in section 3.1.3, the watermark algorithms are used as point of view. Therefore, the embedding-, attacking- and detection/retrieval functions are used for the evaluation. In section 3.1.4 different applications are used as point of view, where different application scenarios and different requirements are used to provide an objective evaluation of watermarking algorithms.

The evaluation of watermarking algorithms is divided into three main classes in the literature, which are described in the following itemization briefly. A detailed description of the classes is introduced in the following subsections.

- A quality depended evaluation is introduced in [LD04]. Thereby, this class specifies and divides the parameters for the watermarking algorithm and/or the attack parameters in a defined range of expected and required quality level of the audio signal. This class divides between high and low quality and between robust and fragile watermarking. If, for example, a robust watermark is embedded with a high embedding strength, which decreases the audio quality and the required quality is not important for the application (e.g. telephony or preview scenario), then the used algorithm is associated to the low quality, robust profile. The following subsection 3.1.2 discusses this class of evaluation methodology in more detail.
• A process depending classification, introduced in [LDLD05], classifies the evaluation itself into three profile classes embedding, attacking and detecting/retrieval profiles. The “life cycle” of a watermark or the watermarking algorithm perspective is the motivation for it. Because as first, the embedding function embed the watermark. Secondly, the marked signal is distributed, stored or used, which is comparable with attacks against the watermark. The third and last phase of the “life cycle” or the watermarking scheme is the detection/retrieval of the embedded message. This classification is introduced in section 3.1.3 in more detail.

• An application depending classification, introduced in [LDLD05, LD06a], classifies the evaluation into three profile classes. But in this classification, the user skill, evaluated watermark property and the selected application scenario are used for watermark evaluation with profiles. The profiles are basic, extended and application profiles. Whereby, the basic profiles, comes from subsection 3.1.1, evaluate only the single properties of a watermarking scheme. In contrast, the application profiles evaluate the watermarking scheme from a special selected application point of view. This class of profiles is introduced in more detail in section 3.1.4.

Derived from the required audio signal modifications, motivated from audio processing and application scenarios, as well as from the different classifications, the following subsections define and classify audio attacks and profiles for digital audio watermark evaluation.

3.1.2 Quality Dependent Benchmarking

Derived from the required or expected audio quality, a new digital audio watermark evaluation methodology of benchmarking with the focus on the expected or required audio quality is defined and discussed in this section.

In the literature [LD04] are evaluation profiles introduced in a term of virtual goods like music or speech. There are different profiles for audio identified, which are describe and classified. The general main classification is in respect to robustness or fragility, and transparency by identifying high or low quality constraints by the applications. Transparency has to be determined after the watermark embedding as well as after the benchmarking attacks. The following description of main profiles based on robustness/fragility and transparency, gives an overview of the classification combined with a short description of the profiles.

**Low quality robust:** Is useful for scenarios, where the robustness of the embedded information is more important than the transparency. Examples are: preview of virtual goods, advertisement, telephone, Internet radio or logging function. For the evaluation process this means, that the evaluation or attack can be executed stronger to attack the digital robust watermark as hard as possible.

**Low quality fragile:** Can be used for scenarios, where the quality is not an essential parameter, but the fragility is important. Examples are the same like for low quality robust: preview of virtual goods, advertisement, telephone or logging function. The degree of fragility can vary depending on the application scenario; therefore sub-profiles become necessary.
**High quality robust:** Reflects the high quality scenarios with an embedded robust audio watermark. Examples are: CD or DVD audio data, cinema application and concert or theatre scenarios. Transparency and robustness are most important parameters. In general modifications caused by watermark embedding or evaluation processes have impacts on the audio signal quality and the transparency has to be determined. If the transparency or the robustness is affected then the benchmarking detects a vulnerability of the watermarking algorithm.

**High quality fragile:** Describes the high quality scenarios for fragile digital audio watermarks. Examples are: CD or DVD audio data, cinema application and concert or theatre scenarios. The embedded information can be used to identify manipulations on the audio content but the transparency of the watermark is very important and has to be ensured. As same as for the low quality fragile profile the degree of fragility can vary depending on the application scenario and sub-profiles become necessary.

The Figure 3.2 shows a visualization of the introduced classification above whereby existing evaluation strategies can be categorized too. For example, the geometric or removal attacks can also be classified into the group of low or high quality results (depending on the parameter sets) as well as they can be used to evaluate robust or fragile digital audio watermarks. This new evaluation methodology is open for many other evaluation and benchmarking strategies, like for example hybrid watermark as combination of robust and fragile digital audio watermarks. The combination of robust and fragile watermarks can be achieved by combining them together or, depending on the used watermarking algorithms, multiples embedding with firstly a robust and secondly a fragile watermark provides are used. The usage of such embedding techniques provides so called *hybrid watermarks*.

![Diagram of Hybrid Watermarks](image)

**Figure 3.2: General Quality Depended Benchmarking**

The description above is introducing the necessity of sub-profiles. In [LD04] are these sub-profiles introduced and assigned to the embedding, detection/retrieval or the attacking function. In the following, a summarization of [LD04] represents the sub-profiles and their affiliation to the embedding, detection/retrieval and attacking function.

The exemplary selected scenarios are used to define sub-profiles with their association to robustness/fragility and their required audio quality as follows:

**Annotation:** Is useful for annotation like combining information with the content. It can be useful for low or high quality robust or fragile watermarks. Examples are: karaoke...
applications, affiliates programs or just additional information. The most important watermark parameter for this sub-profile is to evaluate the capacity to embed enough information into the audio content.

**Key space:** This profile is important for applications where security is important in respect to attacks to the watermarking key. Similar to crypto analysis the key space has to be large and free from weak keys. In this profile, the used key space is evaluated or defined depending on the key length as well as by considering weak keys [FGS04b].

**Collusion resistance:** The importance of this profile can be seen for customer identifications also called fingerprint watermarks and also a sub-profile of security. Due to the nature of using different watermarks on the same or similar content specific attacks called collusion attacks are known and have to be evaluated during benchmarking [DD03, Dit00].

**Long time:** The evaluation with this profile can be important for streaming application scenarios or very large content files. Examples are Internet radio, radio streams or long audio files. Important is that the length of the audio file can be a special case for the embed algorithm, because of some internal variables or loops during the embedding/retrieval process. One first goal of this profile is to evaluate vulnerabilities caused by implementation like coding mistakes causing variables to overflow. A second goal is to determine the security of the watermark for example the watermark period caused by the pseudo random noise generator as general design vulnerability.

**DA/AD:** Digital analogue conversion is usable for broadcast applications, cinema or concerts. The DA/AD conversion – here more specifically as over the air transmission – is on one hand a kind of attack to try to disable or destroy the watermarks and on the other hand a typical usage scenario for each virtual good. The user must perform a conversion to hear the file and benefit from the virtual goods. Many researchers use this kind of scenario to evaluate their robustness of digital robust audio watermarks [MAJ02].

**Hidden communication:** This profile evaluates the security by searching directly for the embedded or hidden message. Statistical analysis e.g. chi-square-test [WP00] or RS-stegoanalyse [FGD01] are used here to perform a decision about possible embedded information.

**Calculation time:** An essential parameter of the profile is the performance of the embedding and detection rate by indicating ranges for embedding and detection time frames. Thereby, this profile evaluates the embedding, retrieval or evaluation time (speed), which is interesting, for example, for real time applications or when an algorithm is developed in hardware as a watermarking or evaluation device.

**Lossy compression rates:** The evaluation of a digital audio watermarking scheme, with this profile determines the resistance or fragility to specific encoder models and to specific compression rates. Here, the profile supports actual known audio encoder models like MP3, OGG, WMA or VQF [Gri02, Xip, MAJ02]. The following audio compression ratios can be selected for evaluation: 8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 144, 160, 192, 224, 256, and 320 kbps as fixed bit rates or variable bit rates with quality steps or minimum and maximum bit rates. Depending on the used encoder model different fixed and variable bit rates can be used.
**Degree of fragility:** As already indicated in the low and high quality fragile profile definition the degree of fragility can vary in different applications. Therefore this sub-profiles allows to determine this parameter in more detail. The idea is to allow the user to specify three categories. The first one is highly fragile and no bit change is allowed. The second choice is a semi-fragile category where the user can select from a set of single or combined attacks from SMBA. The third is the content-fragile category where all content-preserving transformations known from SMBA are selected.

The affiliation of the main- and sub-profiles is shown in the Table 3.3. Thereby, the symbol “●” indicated the affiliation between the profiles and the embedding, detection/retrieval and evaluation/attacking functions.

<table>
<thead>
<tr>
<th></th>
<th>Embedding, Detection/Retrieval</th>
<th>Evaluation, Attacking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low quality robust</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Low quality fragile</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>High quality robust</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>High quality fragile</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Annotation</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Key space</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Collusion resistance</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Long time</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>DA/AD</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Hidden communication</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Calculation Time</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Lossy compression rates</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Degree of fragility</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Table 3.3: Profiles and their Affiliation

The introduced sub-profiles are used for evaluation and their formal definitions are introduced in section 3.1.5.

### 3.1.3 Watermarking Life–Cycle Phases

The defined evaluation methodology is open for different points of view. In this section, our developed methodology of benchmarking with the focus on the watermarking processes are defined and described as new profile based evaluation approach. The idea behind is the typical digital audio watermarking sequence of functions, whereby, as fist, an audio test set is chosen, secondly the watermark is embedded, thirdly an attack occurred and finally, the watermark is detected or retrieved. Therefore, the complete process commencing with the original unmarked audio signal up to the detecting process is mapped to the benchmarking and evaluation concept to provide a function based evaluation.

In general, the usage of digital watermarking can be simplified as follows. An unmarked
(mostly original) signal \((S, \text{ with } S \in \mathbb{S})\) is the source signal, where the watermark \((w)\) is embedded by using an embedding function \(E\). The result is the marked signal \(S_E\). It can be defined, that this process is done in a secure environment. The following step could be, for example, the distribution of \(S_E\) over the Internet or storage of it to provide authenticity or integrity checks. These processes can be seen as an insecure part, where attacks \((A_{i,j} \in A)\) occur on \(S_E\), for example, performed by SMBA. After distribution of \(S_E\), the signal is defined as \(S_{EA}\) because potential attacks could have destroyed the watermark. A detecting function \(D\) tries to detect the watermark \(w\) or a retrieval function \(R\) tries to retrieve the embedded message \(m'\). The detection/retrieval can be done in a secure or insecure environment, depending on the used application of the watermarking algorithm.

The complete introduced scenario is new defined in this work as life cycle of a watermark, because it begins with embedding and ends with detection/retrieval. The following Figure 3.3 introduces this life cycle and shows, where the secure and insecure parts are expected.

![Figure 3.3: General Embedding-, Attacking- and Detecting Functions](image)

The usage of a watermarking algorithm and the classification from the audio signal point of view in the above summarized embedding \(E\), attacking \(A\) and detection \(D\), retrieval \(R\) functions is the motivation for an evaluation with the watermarking algorithm perspective. Therefore, the three main functions\(^2\) \((E, A \text{ and } D,R)\) are used to provide a classification of evaluation processes. Each of these three main functions can be split into sub-evaluation functions. For example, the attacking function can be one single attack or a sequence of attacks. But the sub-functions are not in focus, only the main processes are analyzed and discussed to provide a watermarking algorithm perspective evaluation.

The evaluation can be done in several different ways. One possibility is, to evaluate each of the main functions separately. Therefore, the evaluation is classified into embedding, attacking and detecting profiles [LDLD05]. Furthermore, two main functions, for example, \(E\) and \(A\) or all three can be combined for the evaluation to provide a more global evaluation perspective. The following Figure 3.4 shows it.

\(^2\)There are three main functions, because the detection and retrieval function \(D\) and \(R\) are seen as one function
Furthermore, Figure 3.4 introduces the assignment of the profiles to the evaluation processes, whereby the profile notation is as follows. The profiles are defined as follows: an embedding profile \( P_{E} \) is noted as \( P_{E} \), an attacking profile \( P_{A} \) is noted as \( P_{A} \) and a detection/retrieval profile as \( P_{D} \). There are also combinations possible like \( P_{E/A} \), \( P_{E/A/D} \) or \( P_{E/D} \). The profile internals measure the requested watermarking property with the defined measurements \( \text{tra}_E, \text{com}_E, \ldots \) for the embedding function, \( \text{tra}_A, \text{tra}_A, \text{com}_A, \text{rob}, \ldots \) for the attacking function and \( \text{com}_D, \text{com}_R, \text{det}, \ldots \) for the detection/retrieval function. The assignment of the exemplary sub-profiles from section 3.1.2 to the new introduced watermarking algorithm evaluation perspective is shown in the following Figure 3.5.

The introduced selected sub-profiles, derived from the watermarking life cycle, are used for evaluation and their formal definitions are introduced in section 3.1.5.

### 3.1.4 Benchmarking from the Application Point of View

As already described, the methodology is open to provide an evaluation of digital audio watermarking with different point of views. In this section, the methodology of benchmarking with the focus on the application scenarios is described. Therefore, different abstraction levels of application are defined, discussed and used in this work to provide another classification of the evaluation profiles. With these classes, it is able to design evaluation profiles with different point of views. Here, it is able to distinguish between watermark designer, which have a deep inside knowledge and end users, with few inside knowledge.

If an evaluation of watermarking schemes from the point of end user, or application scenario is needed, then the user has specific design requirements to the watermarking algorithm. For example, the user has as specific criteria, a needed quality level of the audio signal, a specific information, which must be embeddable or a required property of the watermarking scheme. Therefore, the user can have a decision about single properties or parameters or the working domain, which is used for embedding or the user does not care about such details. In the last case, the end user wants to know, if a digital audio watermarking scheme can be used for a given application scenario or not, and the watermark evaluation system gives...
recommendations to support the end user decision. Rather, an application scenario is fixed and the evaluation with the user requirements should help to identify possible watermark algorithms and its parameters.

The other case of users is, for example, the watermark algorithm researcher, designer and developer. From their point of view, the elementary properties (see section 3.1.1) are very important and are should be tuned. Therefore, the users are interested in the elementary, basic properties of a watermarking algorithm.

From this motivation, a classification from the application point of view into basic, extended and application profiles is provided [LD06a]. The following Figure 3.6 introduces this classification in basic, extended and application profiles. Thereby, the basic profiles evaluate the elementary properties of a watermarking scheme (like, transparency or capacity). In contrast, the application profiles evaluate a watermark algorithm with the required properties of the application itself. Thereby the requirement of the application provides the evaluation parameters. The extended profiles are small parts of an application and much more complex as the basic profiles.

Thereby, the details and properties of the digital watermarking scheme are in focus of the evaluation, by choosing the basic profiles, whereby the level of abstractness increases if extended profiles are selected. If the application profiles are used for the watermarking algorithm evaluation, then the evaluation process focuses on a specific application scenario. The abstractness of evaluation details and the knowledge about the watermarking algorithm
increases from the basic to the application profiles and seems to be unimportant from the view of application.

The introduced sub-profiles are used for evaluation and their formal definitions are introduced in section 3.1.5

### 3.1.5 Benchmarking Metrics for the Profile Based Approach

In this section the benchmarking metrics and the evaluation profiles are formalized and their measurements methods and parameters are introduced. The profiles are classified into the perspective of watermarking algorithms as described in section 3.1.3 to show the usage of the evaluation methodology. Note, the following definitions can also be used to classify the evaluation profiles into their application levels as described in section 3.1.4, which is not in focus of this work.

A few publications in the literature introduce the evaluation of watermarking algorithms by using profiles instead of single attacks [LDLD05, LD04, LDSV05]. The profile evaluation divides between quality levels (high and low quality) [LD04], embedding, attacking and detecting/retrieving profiles [LDLD05] and in basic, extended and application profiles [LD06a]. The notation which describes the profiles is determined in the literature [LDSV05] and briefly introduced in section 3.1.3. In the following, the notation for the profile description used in...
this report is summarized and introduced.

The formal description of the profiles depends on the point of view. If, for example, the watermark life-cycle phases perspective (see section 3.1.3) is in focus, then the notation is as follows: embedding profiles: \( P_{E-name} \), attacking profiles: \( P_{A-name} \) and detection/retrieval profiles: \( P_{D-name} \). Whereby the \textit{name} identifies the used profile by noting the profile name there. This concept is used for all existing profiles.

Depending on the evaluated watermarking function (embedding, detection/retrieval or attacking) the profile belongs to an embedding profile if the function embeds the watermark or it belongs to an attacking profile, if the function attacks the watermark. In general, the profiles are divided into these 3 classes, which are presented and described below with their formal description, coming from [LDLD05]. In the following, the basic, extended and application profiles are described.

Constitutive on section 3.1.3 and 3.1.4, profiles are defined and introduced in the following. Firstly, the basic profiles, secondly the extended profiles and last the application profiles are described and introduced. It is noted, that the introduced profiles are not a complete, entire description of all existing profiles. Rather there exist more profiles, especially for the application scenarios. Therefore, the introductions should motivate the usage of evaluation with profiles and provide a general concept to design and implement new profiles.

The description below introduces the profiles and the required audio signals. The global parameters of the profiles are as follows. All profiles have the three parameters input signal (in–signal), output signal (out–signal) and additional parameters (param). These three parameters and the profile specific addition parameters, defines within param, are concatenated by the symbol ‘\( || \)’ when the profile is used for watermark evaluation. The general profile definition is shown in the following equations, whereby exemplary the embedding profile capacity and embedding, attacking profile transparency are selected.

\[
P_{E-Capacity}(in–signal \ || \ out–signal \ || \ param) \tag{3.95}
\]
\[
P_{E/A-Transparency}(in–signal \ || \ out–signal \ || \ param) \tag{3.96}
\]

The parameter in–signal defines the audio signal, which is used to work with it from the profile and its depending function. It can be the original audio signal \( S \), the marked audio signal \( S_E \) or the marked and attacked audio signal \( S_{EA} \). The coherence between the profile description and the corresponding input signal in–signal is in the following Table 3.4 with the exemplary profile name \textit{name} shown. There exist exceptions, where no audio signal is used as input signal. These exceptions are also discussed below.
<table>
<thead>
<tr>
<th>Input Parameter (in–signal)</th>
<th>Profile</th>
<th>Assigned Input Audio Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>embedding profile</td>
<td>$P_E$–name</td>
<td>$S$</td>
</tr>
<tr>
<td>attacking profile</td>
<td>$P_A$–name</td>
<td>$S_E$</td>
</tr>
<tr>
<td>detection/retrieval profile</td>
<td>$P_D$–name</td>
<td>$S_{EA}$</td>
</tr>
<tr>
<td>embedding-, attacking profile</td>
<td>$P_{E/A}$–name</td>
<td>$S, S_E$</td>
</tr>
<tr>
<td>embedding-, attacking- and detecting Profile</td>
<td>$P_{E/A/D}$–name</td>
<td>$S, S_E, S_{EA}$</td>
</tr>
</tbody>
</table>

Table 3.4: Input Signal of the Profiles

The parameter out–signal, which is also defined for all profiles specifies the audio signal of the profile, which depends on the input signal and its modification performed by the associated function. There detection/retrieval profiles do not provide an output audio signal. Therefore, for these profiles out–signal is empty ($\emptyset$). The following Table 3.5 shows the coherence of the output signal (out–signal) of the profiles and the associated audio signal.

<table>
<thead>
<tr>
<th>Output Parameter (out–signal)</th>
<th>Profile</th>
<th>Assigned Output Audio Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>embedding profile</td>
<td>$P_E$–name</td>
<td>$S_E$</td>
</tr>
<tr>
<td>attacking profile</td>
<td>$P_A$–name</td>
<td>$S_{EA}$</td>
</tr>
<tr>
<td>detecting profile</td>
<td>$P_D$–name</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>embedding-, attacking profile</td>
<td>$P_{E/A}$–name</td>
<td>$S_E, S_{EA}$</td>
</tr>
<tr>
<td>embedding-, attacking- and detecting Profile</td>
<td>$P_{E/A/D}$–name</td>
<td>$S_E, S_{EA}$</td>
</tr>
</tbody>
</table>

Table 3.5: Output Signal of the Profiles

The parameter param defines the required parameters of the profile and are introduced for each profile separately in the following subsections.

3.1.5.1 Definition of Basic Profiles

For watermark designers and developers the watermark properties and the impact of the watermark parameters or audio content are often interesting. Therefore, in this subsection the basic profiles, which evaluate and measure the watermark properties, and their required parameters are defined and introduced. Derived from the methodology from section 3.1.3 the basic profiles are classified into embedding, attack and/or detection/retrieval profiles. The following description uses the defined terminology from subsection 3.1.1 for the profiles. Furthermore, the profiles, which are briefly introduced in [LD06a] are enhanced to improve their usability.

Robustness/Fragility: Evaluates the robustness of an embedded digital audio watermark by its detect ability and/or retrieve ability after one or a sequence of malicious single
attacks or non-malicious signal processing operations. The definition of robustness from page 20 is used for measurement. This profile is an attacking profile and defined as:

$$P_{A-Robustness} (\text{in–signal} \ || \ \text{out–signal} \ || \ \text{param})$$  \hspace{1cm} (3.97)
$$\text{param} = (\text{wm-alg} \ || \ \text{wm-opt} \ || \ \text{at-alg} \ || \ \text{at-opt} \ || \ \text{add-opt})$$  \hspace{1cm} (3.98)

The parameter “wm-alg” defines the watermarking scheme, from which the robustness is evaluated. In addition, the parameter “wm-opt” defines the parameters required from the algorithm wm-alg, “at-alg” defines the attacking function with its parameters defines in “at-opt” to attack the embedded watermark. The last parameter “add-opt” defines additional options for the profile, like a threshold.

The robustness itself is measured by computing the values of relative robustness $\text{rob}_{rel}$, average robustness $\text{rob}_{ave}$ or the minimal robustness $\text{rob}_{min}$.

**Transparency:** Evaluates the audible distortion due to signal modifications occurred by watermark embedding or attacking. Therefore, this profile is classified as an embedding and attacking profile depending on its usage. It is defined as:

$$P_{E/A-Transparency} (\text{in–signal} \ || \ \text{out–signal} \ || \ \text{param})$$  \hspace{1cm} (3.99)
$$\text{param} = (\text{alg} \ || \ \text{alg-opt})$$  \hspace{1cm} (3.100)

In Equation (3.99) and 3.100, the parameter in–signal is always an audio signal, from which the transparency is measured. The parameter out–signal is the resulting audio signal. The parameter “alg” defines the used function (embedding or attacking) with its needed parameters defined in “alg-opt”. If this profile is used as embedding profile, then the parameter alg identifies the used watermarking embedding process ($E(S, k, w)$), which is introduced in the following equation.

$$P_{E-Transparency} (S \ || \ S_{E} \ || \ \text{param});$$  \hspace{1cm} (3.101)
$$\text{param} = (E \ || \ p_{E})$$  \hspace{1cm} (3.102)

In this case, the transparency measure method is based on the predefined relative embedding transparency $\text{tra}_{E,rel}$ or the associated average $\text{tra}_{E,ave}$, minimum $\text{tra}_{E,min}$ or maximum $\text{tra}_{E,max}$ values. If this profile is an attacking profile, then the parameter alg identifies an attack (or concatenations of attacks), from which the transparency is measured (introduced in the following equation).

$$P_{A-Transparency} (S \ || \ S_{EA} \ || \ \text{param})$$  \hspace{1cm} (3.103)
$$\text{param} = (A_{ij} \ || \ p_{A})$$  \hspace{1cm} (3.104)

In this case, the transparency measure method is based on the predefined relative attacking transparency $\text{tra}_{A,rel}$ or the associated average $\text{tra}_{A,ave}$, minimum $\text{tra}_{A,min}$ or maximum $\text{tra}_{A,max}$ values. It is noted, that the attacking transparency is also defined between the original audio signal $S$ and the marked, attacked audio signal $S_{EA}$. In this case, the Equation (3.103) differs in their input signals and is defined as follows.

$$P_{A-Transparency}^{*} (S \ || \ S_{EA} \ || \ \text{param})$$  \hspace{1cm} (3.105)

Whereby the transparency measure method is based on $\text{tra}_{A,rel}^{*}$, $\text{tra}_{A,ave}^{*}$, $\text{tra}_{A,min}^{*}$ and $\text{tra}_{A,max}^{*}$.
**Capacity:** Evaluates the amount of possible embedding information into the audio signal. Depending on the usage, it is an embedding, attacking or detection profile. As embedding profile, the amount of data, which should be embedded (\(c^E\)) is measured. If it is a detection profile, then the retrieved capacity of the retrieval function \(R\) directly after embedding is measured. In contrast, if it is an attacking profile, then the retrieved capacity of the embedded message for \(n\)-bit watermarking schemes after performing an attack (or a sequence of attacks) is measured. The profile is defined as:

\[
P_{E/A/D-Capacity}(\text{in–signal} \parallel \text{out–signal} \parallel \text{param}) = (\text{alg} \parallel \text{alg-opt})
\]

The parameter in–signal is always an audio signal used for capacity measurement. The parameter out–signal is the resulting output audio signal. The parameter “alg” defines the used function (embedding or attacking) with its needed parameters defined in “alg-opt”. If it is an embedding profile, then the parameter “alg” identifies the embedding function.

\[
P_{E-Capacity}(S \parallel SE \parallel \text{param}) = (E \parallel PE)
\]

In this case, the capacity measure method is based on the predefined relative embedding capacity \(c^E_{rel}\). Thereby, the message is embedded and the detection/retrieval function measures, if the complete message fits into the audio signal \(S\).

If the profile is used as detection profile, then the retrieved capacity directly after embedding is measured (useful for \(n\)-bit watermarking schemes).

\[
P_{D-Capacity}(S \parallel SE \parallel \text{param}) = (R \parallel PR)
\]

In this case, the capacity measure method is based on the predefined relative retrieving capacity \(c^R_{rel}\) (or \(c^R_{rel}\)). Based on these retrieving capacities, the average \(c^{Rave}\), minimum \(c^{Rmin}\) and maximum \(c^{Rmax}\) retrieving capacity can be measured.

If the profile is used as an attacking profile, then the retrieved capacity is measured after performing attacks.

\[
P_{A-Capacity}(S_E \parallel SE_A \parallel \text{param}) = (A_{i,j} \parallel PA)
\]

In this case, the capacity measure method is based on the predefined relative attacking capacity \(c^{Arel}\) (or \(c^{Arel}\)). Based on these attacking capacities, the average \(c^{Aave}\), minimum \(c^{Amin}\) and maximum \(c^{Amax}\) attacking capacity can be measured.

**Complexity** evaluates the complexity of an embedding, detection/retrieval or attacking function and is defined as embedding, attacking and detection profile as follows.

\[
P_{E/A/D-Complexity}(\text{in–signal} \parallel \text{out–signal} \parallel \text{param}) = (\text{alg} \parallel \text{alg-opt})
\]
The parameter in–signal is an audio signal used for complexity measurement. The parameter out–signal is the resulting output audio signal. The parameter “alg” defines the used function (embedding, attacking or detection/retrieval) with its needed parameters defined in “alg-opt”. In the following is exemplary the embedding function, attacking function and detection/retrieval function shown.

\[
P_{E-\text{Complexity}}(S \parallel S_E \parallel \text{param}) = (E \parallel p_E)
\]
\[
P_{A-\text{Complexity}}(S_E \parallel S_{EA} \parallel \text{param}) = (A_{i,j} \parallel p_A)
\]
\[
P_{D-\text{Complexity}}((S_E, S_{EA}) \parallel \text{result} \parallel \text{param}) = (\{D, R\} \parallel \{p_D, p_R\})
\]

The complexity is measured, by using the predefined complexity measure function \(C(F)\) from page 14. With it, the relative embedding complexity \(\text{com}_{E,\text{rel}}\), relative attacking complexity \(\text{com}_{A,\text{rel}}\) and relative detection/retrieval complexity \(\text{com}_{D,\text{rel}}, \text{com}_{R,\text{rel}}\) and all derived average, minimum and maximum complexities can be measured.

**Verification:** This profile provides information about the watermarking algorithms and its type of verification, public (blind), informed and private (non-blind). It depends on the watermarking embedding function and therefore, it is an embedding profile defines as follows.

\[
P_{E-\text{Verification}}(\text{in–signal} \parallel \text{out–signal} \parallel \text{param}) = (\text{wm-alg} \parallel \text{wm-opt})
\]

The parameter in–signal is an audio signal used for complexity measurement. The parameter out–signal is the output audio signal. The parameter “wm-alg” defines the used watermarking scheme with its needed parameters defined in “wm-opt”. The verification measure is based on Equation (3.82), which measures the verification ver.

**Security** evaluates the security of a watermarking algorithm. Thereby, a specific security, like the resistance against collusion attacks, cryptographic or protocol attacks, a brute force key search or key space reduction [KVH00, MHJSM06] or specific cases of an encrypted embedding message are in focus of this profile. Derived from the security definition in Equation (3.91), the profile is an embedding and attacking profile and the total security can be measured as follows:

\[
P_{E/A-\text{Security}}(\text{in–signal} \parallel \text{out–signal} \parallel \text{param}) = (\text{wm-algo} \parallel \text{wm-opt} \parallel \text{sec}_{\text{tot}})
\]

Whereby, in–signal and out–signal are the input and output audio signal and wm-algo defines the evaluated watermark scheme with its parameters defined in wm-opt. In case, that only one specific security property should be analyzed, then not this specific
watermark security is measured. For example, the collusion resistance, which is an attacking profile:

\[ P_{A-\text{Security}_{\text{collusion}}} \left( \{ \text{in-signal}_1, \ldots, \text{in-signal}_n \} \parallel \text{out-signal} \parallel \text{param} \right) \]  \hspace{1cm} (3.126)
\[ \text{param} = (\text{wm-algo} \parallel \text{wm-opt} \parallel \text{sec}_{\text{coll}}) \]  \hspace{1cm} (3.127)

The parameters in-signal\(_1\) to in-signal\(_n\) with \( n \in \mathbb{N} \) and \( n \) is the total number of audio signals, the collusion security \( \text{sec}_{\text{coll}} \) is measured by performing \( n \) collusion attacks and the corresponding output audio signal out-signal is calculated. How the collusion attack is performed, depends on the definition of \( \text{sec}_{\text{coll}} \), which is introduced on page 27.

**Invertibility** provides information about the watermarking algorithms and its possibility of inversion predefined on page 26. The profile is an embedding profile and defined as follows.

\[ P_{E-\text{Invertibility}}(\text{in-signal} \parallel \text{out-signal} \parallel \text{param}) \]  \hspace{1cm} (3.128)
\[ \text{param} = (\text{wm-alg} \parallel \text{wm-opt}) \]  \hspace{1cm} (3.129)

The parameter in-signal is an audio signal used for invertibility measurement. The parameter out-signal is the output audio signal. The parameter “wm-alg” defines the used watermarking scheme with its required parameters defined in “wm-opt”. The invertibility measure is based on Equation (3.81), which measures the invertibility inv.

The following Table 3.6 summarizes the basic profiles and shows their association to the embedding, attacking and detection/retrieval function. Thereby, most basic profiles focus on the embedding and attacking functions, whereby only two basic profiles measure a property of the detection/retrieval function.

<table>
<thead>
<tr>
<th>Basic Profile</th>
<th>Embedding Function</th>
<th>Attacking Function</th>
<th>Detection/Retrieval Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{A-\text{Robustness}} )</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>( P_{E/A-\text{Transparency}} )</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>( P_{E/A-\text{Capacity}} )</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>( P_{E/A/D-\text{Complexity}} )</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>( P_{E-\text{Verification}} )</td>
<td>•</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_{E/A-\text{Security}} )</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>( P_{E-\text{Invertibility}} )</td>
<td>•</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.6: Summarization of Basic Profiles and its Classification into Embedding, Attacking and/or Detection/Retrieval Profiles

### 3.1.5.2 Definition of Extended Profiles

Basic profiles focus on the evaluation and on the measurement of the fundamental watermark properties depending on the watermark properties or the audio content. The impact of
complex scenarios (not application scenarios) or evaluation parts used in application profiles are neglected. Therefore, in this subsection exemplary selected extended profiles and their required parameter sets are defined and introduced. Thereby, the extended profiles are like the basic profiles classified into embedding, attack and/or detection/retrieval profiles. Note, the most common and often used extended profiles are chosen to introduce and motivate the profile based evaluation with them. Beside the introduced extended profiles many other extended profiles exists, which are not introduced here.

Extended profiles are more complex and do not measure a single property of the watermarking algorithms (like the basic profiles). In general, the extended profiles perform a single part of a real world scenario on the watermarked audio signal, which can be a part of an complex application scenario. The following description introduced exemplary selected extended profiles, which are often used as part for practical audio application scenarios, predefined in [LD06a] to show the usage of the methodology. Note, there exists many other extended profiles and the entire

**Annotation:** Annotation watermarking (sometimes also called caption watermarking or illustration watermarking) is used to embed supplementary information directly in the media, so that the additional information is directly integrated and cannot get easily lost (e.g. meta data like the audio description fields). Today, a wide range of applications for annotation watermarking can be found, especially also to watermark specific media objects like song sequences. As discussed in [VD05] in comparison to copyright watermarking we do not expect any dedicated removal, cryptographic or protocols attacks in general. As the annotated data would lose value in most cases there is no attack motivation and most security properties of the watermarking scheme have a minor relevance. Even only a limited set of expected geometric attacks play an important role, like robustness against add noise, cutting and compression seems to be the most the important one. Highly of interest are the parameters capacity and transparency. It is an embedding profile and defined as follows:

$$P_{E-\text{Annotation}}(\text{in–signal} || \text{out–signal} || \text{param})$$

$$\text{param} = (\text{alg} || \text{alg-opt} || \text{tra} || \text{cap})$$

The parameter “in–signal” is the input audio signal and the parameter “out–signal” defines the resulting output audio signal. The parameter “alg” defines the watermarking scheme, which is evaluated with its parameters defined with “alg-opt”. To define the lowest acceptable quality degree of the marked signal (embedding transparency), the parameter “tra” specifies it. The required embedding capacity can be defined with the parameter “cap”. The evaluation with this extended profile can be split into the evaluation with the two basic profiles $P_{E/A-\text{Transparency}}$ and $P_{E/A-\text{Capacity}}$. The transparency measure answers the question, if there are noticeable artefacts in the marked audio signal, which the user will probably not accept for professional and semi-professional purposes like publishing and presentation? However, the capacity measurement identifies how many bits can be spread over the whole audio signal or directly stored in the related parts required for the annotation.

**Calculation Time:** The profile internals are similar to the basic profile complexity defined above ($P_{E/A/D-\text{Complexity}}$). The difference between both is, that the profile Calculation
Time measures the complete and overall complexity (like many processes or including IO operations) needed in a framework instead for a single process. It is an embedding, attacking and detection profile and defined as follows.

\[ P_{E/A/D-Calculation_{Time}}(\text{in–signal} \ || \ \text{out–signal} \ || \ \text{param}) \quad (3.132) \]

\[ \text{param} = (\text{alg} \ || \ \text{alg-opt}) \quad (3.133) \]

The parameter “in–signal” is an audio signal used for measuring. The parameter “out–signal” is resulting output audio signal. The parameter “alg” defines the used function or process (embedding, attacking or detection/retrieval with its needed additional processes like IO, etc.) with its needed parameters defined in “alg-opt”.

**Combined Algorithms:** If this profile is selected for watermark evaluation, the extended profile embeds watermark information by using two or more different embedding functions using either the same watermarking algorithm with different embedding parameters or different watermarking algorithms. It is an embedding and detection profile and defined as follows:

\[ P_{E/D-Combined_{Algorithms}}(\text{in–signal} \ || \ \text{out–signal} \ || \ \text{param}) \quad (3.134) \]

\[ \text{param} = (\{\text{alg}_1, \text{alg}_2, \ldots, \text{alg}_n\} \ || \ \{\text{alg-opt}_1, \text{alg-opt}_2, \ldots, \text{alg-opt}_n\}) \quad (3.135) \]

Whereby “in–signal” and “out–signal” are the corresponding input and output audio signals. The parameter “alg\_1” defines the first used embedding (detection/retrieval) function, whereby \( n \) defines the number of used embedding functions. The parameters “alg-opt” are the corresponding parameters needed from the embedding or detection/retrieval functions.

**DA/AD** evaluates the robustness of a watermarking algorithm against digital-analogue and analogous-digital conversion. It is an attacking profile and defined as follows.

\[ P_{A-DA/AD}(\text{in–signal} \ || \ \text{out–signal} \ || \ \text{param}) \quad (3.136) \]

\[ \text{param} = (\{A_{i,1}, A_{i,2}, \ldots, A_{i,n}\} \ || \ \{p_{A1}, p_{A2}, \ldots, p_{Aj}\}) \quad (3.137) \]

Whereby “in–signal” and “out–signal” are the input and output audio signals. \( A_{i,1}, \ldots, A_{i,n} \) are single attacks provided by SMBA, which have to run in order \( i = 1 \ldots n, \ n+++ \) with the using of single attacks parameters \( p_{A1}, \ldots, p_{An} \). \( i \) defines the type of attacks.

**Estimation Attacks:** Estimation attacks are introduced in [VPP+01] and the concept is based on the assumption to estimate the original data or watermark itself. This extended profile uses invertible single attacks to modify the marked audio signal in that way, that an attack and its inverted attack is performed to get a similar audio signal. It is an attacking profile and defined as follows.

\[ P_{A-Estimation_{Attacks}}(\text{in–signal} \ || \ \text{out–signal} \ || \ \text{param}) \quad (3.138) \]

\[ \text{param} = (A_{i,j} \ || \ p_A) \quad (3.139) \]

Whereby “in–signal” and “out–signal” are the input and output audio signals of this profile. As attack \( A_{i,j} \) with its parameters \( p_A \) are only single attacks possible, which
are invertible. After performing this extended profile against the embedded watermark, the output audio signal “out–signal” is similar to the input audio signal “in–signal”, whereby minor changes, depending on the attack $A_{i,j}$ occurred.

It is noted, that the attack $A_{i,j}$ in this profile is only available, if an inversion of it ($A_{i,j}^{-1}$) exists. Then, this extended profile can be seen as follows.

$$\text{out–signal} = A_{i,j}^{-1}(A_{i,j}(\text{in–signal})) \quad (3.140)$$

**Format conversion:** The idea behind this extended profile is, to convert a given audio signal into an audio signal with different properties. It means that the sample rate, quantization and/or number of channels can be changed. It is part of a real world scenario, where the audio format changes or a format conversion occurred. It is an attacking profile and defined as follows.

$$P_{\text{A–Format Conversion}}(\text{in–signal} || \text{out–signal} || \text{param}) \quad (3.141)$$

$$\text{param} = (f_{\text{SR}} || \text{quant} || \text{channel}) \quad (3.142)$$

Whereby “in–signal” and “out–signal” are the input audio signal and the converted output audio signals of this profile. The parameter $f_{\text{SR}}$ defines the new sample frequency, “quant” the new quantization and “channel” the number of channels of the output audio signal “out–signal”.

**Key Space:** The idea of this extended profile comes from [FGS04b], where a methodology is introduced to identify the used secret key in key depending steganographic schemes. The concept uses exhaustive searches, which are looking for some recognizable structures. In [FGS04b] are methodologies and test results for selected steganographic schemes introduced, which are the motivation to design this profile by using these methodologies and adapting and enhancing them for digital watermarking schemes. It is an attacking profile and defined as follows.

$$P_{\text{A–Key Space}}(\text{in–signal} || \text{out–signal} || \text{alg}) \quad (3.143)$$

The parameter “in–signal” defines the original (marked) audio signal and the parameter “out–signal” defines the corresponding output audio signal. The parameter “alg” defines the possible used algorithm for embedding. As introduced in [FGS04b], with the knowledge about the used embedding function, the profile tries to reduce the key space and to identify possible used secret keys.

**Long Time:** The extended profile evaluates the embedding and detection/retrieval function of a watermarking scheme and the attacking function which attacks the watermark with very large content files [LD04]. Important is, that the length of the audio signal can be a special case for the embedding, attacking and detection/retrieval function, because of some internal variables or loops during embedding, attacking and detection/retrieval. One first goal of this profile is to evaluate vulnerabilities caused by implementation like coding mistakes caused variables overflow. A second goal is to determine the security of the watermark for example the watermark period caused by the pseudo random noise generator (PRNG) as general design vulnerability. This profile is classified as an
embedding, attacking and detection/retrieval profile and defined as follows.

\[ P_{E/A/D-Long\_Time}(\text{in–signal} || \text{out–signal} || \text{param}) \]  
\[ \text{param} = (\text{alg} || \text{alg-opt} || \text{time}) \]  

(3.144)  

(3.145)

The parameter “in–signal” defines the input and the parameter “out–signal” the corresponding output audio signal. The parameter “alg” defines the used algorithm (or function), for example, embedding, attacking or detection/retrieval algorithms with its additional parameters defined in “alg-opt”. The parameter “time” defines the length of the audio signal (input and output). Noted, that the audio file formats have a limit as maximum length. For example, the audio file format WAVE [Pol95] uses 4 Bytes to save the number of sample values, stored in the audio file. Depending on the used number of channels, quantization and sample rate, the total length has a defined maximum value. If an audio signal is longer than this length, then the parameter “time” should be set to \(-1\), because an infinite length is defined by using an audio stream for the evaluation.

**Lossy Compression:** This extended profile evaluates the robustness or fragility of an embedded watermark against lossy compression. This profile performs a real lossy compression on the audio signal. It is classified as an attacking profile and defined as follows.

\[ P_{A-Lossy\_Compression}(\text{in–signal} || \text{out–signal} || \text{param}) \]  
\[ \text{param} = (\text{alg} || \text{alg-opt}) \]  

(3.146)  

(3.147)

Whereby the parameter “in–signal” defines the input audio signal and the parameter “out–signal” the lossy compressed and decompressed audio signal. The parameter “alg” defines the used lossy compression algorithm (like MP3 or OGG), and the parameter “alg-opt” defines the needed parameters depending on the lossy compression algorithm. These parameters can be, for example, the data rate\(^3\) or the quality level\(^4\).

**Packet Loss:** This extended profile is needed to simulates the transmission of audio data over a network where packet loss occurs. Such effects occurs for example, if an audio signal is streamed over the Internet (like voice over IP – VoIP). It is classified as an attacking profile and defines as follows.

\[ P_{A-Packet\_Loss}(\text{in–signal} || \text{out–signal} || \text{param}) \]  
\[ \text{param} = (Remove || RemoveNumber) \]  

(3.148)  

(3.149)

The input audio signal “in–signal” specifies the original (marked) audio signal and the output audio signal “out–signal” is modified in that way, that samples are removed. The additional parameter “param” describes the commonness and size of a packet loss which is simulated. Thereby, the parameter “Remove” defines the distance between the packet loss occurred and “RemoveNumber” defines the number of removed audio samples. This profile can be performed by using the single attack “CutSample” provided by SMBA.

---

\(^3\)Typical data rates are 8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 144, 160, 192, 224, 256, 320 kbits per second

\(^4\)The OGG algorithm has a quality level as parameter from -1 (very low) up to 10 (very high).
It is noted again, that the introduced extended profiles above are exemplary selected extended profiles and should motivate to design new extended profiles needed for the watermark evaluation.

The following Table 3.7 summarizes the exemplary selected extended profiles and shows their association to the embedding, attacking and detection/retrieval functions. Thereby, it is shown, that extended profiles can evaluate one or more watermarking functions with typical malicious or non-malicious operations on the audio signal.

<table>
<thead>
<tr>
<th>Extended Profile</th>
<th>Embedding Function</th>
<th>Attacking Function</th>
<th>Detection/Retrieval Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE−Annotation</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PE/A/D−Calculation_Time</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>PE/D−Combined_Algorithms</td>
<td>●</td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>PA−DA/AD</td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>PA−Estimation_Attacks</td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>PA−Format_Conversion</td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>PA−Key_Space</td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>PA−Long_Time</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>PA−Lossy_Compression</td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>PA−Packet_Loss</td>
<td></td>
<td>●</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.7: Summarization of Exemplary Extended Profiles and their Association to Embedding, Attacking and/or Detection/Retrieval Functions

### 3.1.5.3 Definition of Application Profiles

In this subsection exemplary selected application profiles and their required parameter sets are new defined and discussed. If the evaluation of digital audio watermarking schemes is done from the application point of view, as introduced in section 3.1.3, then application profiles are chosen. They reflect real world scenarios and can be seen as a complete real existing application which used digital watermarking schemes. The usage and therefore, the specific application scenario is used to evaluate a given watermarking scheme in the context of the selected application primary used from end users, with few inside knowledge but open for watermark designer. The idea of application oriented evaluation is briefly introduced in [LD06a]. The result of all application profiles is a recommendation of the evaluated digital watermarking schemes if they are useable with the given parameter and audio sets for the application scenario or not. The application profiles are always classified as attack profile and include the evaluation with one or more basic or extended profiles.

The following description introduces exemplary selected application profiles, predefined in [LD06a] and shows the usage of them. Thereby, a detailed description of the application profile internals is presented for the two exemplary selected application profiles “biometric user authentication” and “perceptual hashing” to show the usage of the methodology. It is also noted, that this description should motivate the usage and evaluation of digital watermarking.
schemes with application profiles and to enhance them with different application scenarios.

**Archive:** This application profile simulates storage of marked signals over a very long time. Therefore, it is important to convert the audio signal into a defined audio format and depending on the storage device, errors can occur over the time. Furthermore, the audio signal could be used to proof the authenticity, which means, that only lossless compression is acceptable. It is defines as follows:

\[ P_{A}^{\text{Archive}}(\text{in–signal} || \text{out–signal} || \text{param}) \]  

Thereby, the input audio signal “in–signal” defines the audio signal, which is stored in the archive. The parameter “out–signal” defines the corresponding output audio signal, whereby the effects occurred during storing in the archive occurred. Depending on the used archive, these effects can be, for example, errors in the audio signal, other format of representation of the audio signal and/or all cognizable effects occurred over a very long time saving.

The evaluation with this application profile requires three different criteria defined within the basic and extended profiles, which are as follows.

- **Long Time:** In an archive are audio signals stored, which have a very long play time. For example, original documents needed for embedding and detection/retrieval. Furthermore, the storing over a long time in an archive performs attacks against the audio signal (bits can be changed or parts are not readable correctly). Therefore, the extended profile \( P_{E/A/D}^{\text{Long–Time}} \) and its evaluation results are part of this application profile.

- **Packet Loss:** During storing the audio signals over a long time, the audio signal changes. These changes vary from swapping of single bits to complete unreadable parts of the audio signal. The evaluation test results of the extended profile \( P_{A}^{\text{Packet–Loss}} \) simulate such audio signal distortion effects and are part of this application profile.

- **Robustness:** The watermark, used in an archive application scenario should be robust enough to provide copyright protection and/or fragile enough to provide an integrity check. Therefore, the robustness and/or fragility of the embedded watermark is part of this application profile. These results are provided by the basic profile \( P_{A}^{\text{Robustness}} \).

The results of the basic and extended profiles, which are part of the application profile “Archive”, provide in their summary a recommendation for the evaluated watermark scheme in the application scenario of “Archive”.

**Biometrics:** The application profile “Biometrics” describes exemplary a biometric authentication system. The general concept is to provide access only, if the user has a specific and pre-registered biometrical attribute. Such typical attributes are, for example, iris, fingerprint, active handwriting, or voice. In general, all biometric systems for user verification or identification need two main steps. The first one is the enrollment phase, where the user has to be enrolled, whereby the biometric parameters are measured and significant reference data are stored. The second step is the biometric verification
or identification, whereby the user provides the system the same biometric information (like iris, handwriting, speech, etc.) which was enrolled in the past. The system compares it with the pre-enrolled biometric reference data. If the user has the same biometric characteristics, then she/he gets access otherwise she/he will be rejected.

In [VSL06] a basic approach to combine biometric with watermarking on the example of speech (audio) is introduced, whereby the meta data are embedded into the speakers reference signal. The idea of watermarking biometric reference data is twofold: firstly, watermarking information can contain additional annotations directly included in the reference data (without additional link structure or storage requirements), secondly, watermarking protects originals (helps to authenticate references and degrades original quality).

With this application profile, a watermarking scheme is evaluated by using it in the biometric user authentication process, whereby the biometric reference data are marked and the influence of the embedding function is measured. It is defined as follows.

\[
P_{A-Biometrics}(\text{in–signal} || \text{out–signal} || \text{param})
\]

\[
\text{param} = (\text{bio-alg} || \text{bio-opt})
\]

The parameters for this profile are the original reference audio signal “in–signal”, the marked reference audio signal “out–signal”, the used biometric authentication algorithm “bio-alg” and additional parameters needed from the biometric system “bio-opt”. The result of this profile is the measure transparency of the embedding function, which affects the biometric system and the information, if the watermark message fits into the biometric reference data or not. The evaluation with this application profile requires two different criteria coming from the basic profiles [LDLD05, LD06a]. These criteria are:

- Transparency: This means, that the worse error rates regarding the embedding function are useable for a defined application scenario. The evaluation results with the basic profile \(P_{E-Transparency}\) must be better than an application depending threshold.

Now, the question is how to define transparency for a biometric authentication system. A current value to appraise biometric systems is the Equal Error Rate (EER), which is defined as intersection between the False Match Rate (FMR) and False Non Match Rate (FNMR) curves of a biometric system [Vie05]. The goal for the following definition of transparency measure is the impact of the watermark to the biometric speech performance based on error rates. The \(EER\) is defined in the interval \([0, 1]\). This value changes, if the biometrical reference data changes or if the threshold for user authentication acceptability is modified. The idea is as follows: Let \(EER\) be the measured equal error rate for unmarked reference signals and \(\overline{EER}\) the measured equal error rate with marked reference signals for a biometric system, then it is possible to apply the following definitions to measure the difference between \(EER\) and \(\overline{EER}\) as transparency measure for the biometric system.

\[
B(EER, \overline{EER}) \to [0, 1]
\]
Where $B()$ is the measure function and the result is in the interval $[0,1]$. A 0 identifies the worst case ($EER$ and $\bar{EER}$ are so different, that no association between the original and marked reference data are identified) and a 1 shows the best result ($EER$ and $\bar{EER}$ are identical, which means, that the impact of the embedding function does not change the biometric system). Thereby, a relative distortion (Biometric Difference Grade BDG) measure for a watermarking scheme $\Omega^*$ and a given audio signal $S$ from an audio test set $\mathcal{S}$ as follows:

$$
\text{bdg}_{\text{rel}}(\Omega^*, S) = \begin{cases} 
1 - \frac{|EER - \bar{EER}|}{EER}, & \text{otherwise} \\
0, & \text{if } \left(1 - \frac{|EER - \bar{EER}|}{EER}\right) < 0
\end{cases} \quad (3.154)
$$

A lower $\text{bdg}_{\text{rel}}$ indicates a worse transparency of the embedding function and a result close to 1 indicates, that no significant transparency decreases occurred. The factor $EER$ is used to realize the effect of the original $EER$ to $\text{bdg}_{\text{rel}}$. Thereby, the influence of slightly changes has a slightly impact on $\text{bdg}_{\text{rel}}$ if $EER$ is high. In contrast, if the $EER$ is low, then little changes between the marked and unmarked biometric reference data affects the computed $\text{bdg}_{\text{rel}}$ more.

However, this definition measures the distortion of a given audio signal $S$, which occurred by applying of $\Omega^*$ and a given biometric system. It is usually better to provide some absolute values which are not assigned to a particular audio signal $S$. Therefore, the absolute, minimum and maximum $\text{bdg}$ is defined as follows over a given audio test set $\mathcal{S}$:

- Average transparency:

$$
\text{bdg}_{\text{ave}}(\Omega^*) = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \text{bdg}_{\text{rel}}(\Omega^*, S). \quad (3.155)
$$

- Maximum transparency:

$$
\text{bdg}_{\text{max}}(\Omega^*) = \max_{S \in \mathcal{S}} \{\text{bdg}_{\text{rel}}(\Omega^*, S)\}. \quad (3.156)
$$

- Minimum transparency:

$$
\text{bdg}_{\text{min}}(\Omega^*) = \min_{S \in \mathcal{S}} \{\text{bdg}_{\text{rel}}(\Omega^*, S)\}. \quad (3.157)
$$

The equations above formalize the biometric difference grade for a particular audio signal by computing the relative value. If the average biometric difference grade over a given test set is needed, then the arithmetic mean is computed. Furthermore, the minimum and maximum $\text{bdg}$ values identify the results for the worst and best case.

- Capacity: which means that the required message can be successfully embedded into the audio signals. Therefore, the evaluation results with the basic profile $P_{E-\text{Capacity}}$ must return that the watermarking scheme provides enough marking positions and the complete message fits into the audio signal.
The results of the basic profiles, which are part of the application profile “Biometrics”, provide in their summary a recommendation for the evaluated watermark scheme in the application scenario of Biometrics.

**Broadcast:** The application profile “Broadcast” describes the application of broadcasting (streaming over the Internet) of an audio signal. This scenario describes, for example, Internet radios, which are broadcast stations and stream their audio signal to the users who want to listen the audio content. It is defined as follows.

\[
P_{A-Broadcast}(\text{in–signal} || \text{out–signal} || \text{param})
\]

\[
\text{param} = (\text{comp} || \text{loss} || \text{opt})
\]

The parameters for this profile are the original input audio signal, defined in “in–signal” and the corresponding output audio signal “out–signal”, which is similar to the audio signal received by the clients. The parameter “comp” described the lossy compression parameters, the parameter “loss” defines the occurred packet loss by transmitting over the Internet and the parameter “opt” defines additional parameters used, for example, for long time measurements, transparency measures or robustness measures. The result of this application profile is a recommendation of the evaluated digital audio watermarking schemes, if these are useable in this application scenario or not. The evaluation with this application profile requires the results of different basic and extended profiles which are parts of this application profile and introduced in the following itemization.

- **Lossy Compression:** This property of the application profile “Broadcast” describes a lossy compression of the transmitted audio signal used to reduce the bandwidth of the network connection and transmission. The characteristic of lossy compression and its evaluation of the watermarking scheme is provided by the extended profile \( P_{A-Lossy\_Compression} \). Thereby, the parameter “comp” of this application profile is passed on the extended profile and the results of this extended profile includes the results of robustness measure (basic profile \( P_{A-Robustness} \)) of the watermarking scheme against lossy compression.

- **Packet Loss:** During the transmission of audio signals over the Internet packet loss occurs because of leaks or delays of IP packet transmission. This part of the application is defined and introduced within the extended profile \( P_{A-Packet\_Loss} \). The parameter “loss” is passed on this extended profile.

- **Long Time:** In the application of Broadcasting, the audio signal is transmitted 24 h per day and 7 days per weeks. It means, that the audio signal does not have an end, which means, that no specific play time or length of the audio signal can be defined. For all functions (embedding, detection/retrieval and attacking), the extended profile \( P_{E-Long\_Time} \) can be used for the evaluation of all functions in the long time scenario.

- **Transparency:** The user, which uses the broadcasted audio signal, for example, listen it in within the application of Internet radio, does not want to have or listen a quality degree occurred by the used watermarking scheme. Thereby, the property transparency, evaluated within the basic profile \( P_{E-Transparency} \) is also a property of this application profile Broadcast.
Complexity: The broadcast station does normally not have the complete water-marked audio content (music, speech of news, etc.) available. Furthermore, the watermark is embedded during broadcasting the audio signal. It means, that the complexity of the embedding function must be fast enough to embed the watermark in real time. This requirement on the watermark scheme can be evaluated with the basic profile $P_{E-Complexity}$.

The results of the basic and extended profiles, which are part of the application profile Broadcast, provide in its summary a recommendation for the evaluated watermark scheme in the application scenario of Broadcast.

**Cinema:** This application profile Cinema evaluates a watermarking scheme in the application scenario of cinema or theatre by simulating the properties of a cinema or theatre. A watermarked audio signal can be used, for example, to trace screener, who makes illegal copies of movies in a cinema to share them in peer-to-peer networks. The application profile is defines as follows:

$$P_{A-Cinema}(\text{in–signal} || \text{out–signal} || \text{param}) \quad (3.160)$$

The parameter “in–signal” defines the input audio signal and the corresponding output audio signal defines the parameter “out–signal”. The parameter “param” defines the properties and characteristics of the cinema application used for the evaluation. The main characteristics can be evaluated by different basic and extended profiles, which are introduced in the following itemization.

- **DA/AD:** The users in a cinema have only access to the analogue content by viewing and listening. If a watermark detection/retrieval after, for example, recording must be successfully, then the embedded watermark must be robust against digital-analogue and analogue-digital (DA/AD) conversion. This part of the cinema can be evaluated with the extended profile $P_{A-DA/AD}$.

- **Format Conversion:** Many watermarking algorithms are designed and implemented to embed the watermark into audio signals with one or two audio channels (mono or stereo) or with a specific sampling frequency. To use these watermark schemes in the application scenario of “cinema”, then the watermarking scheme must be redesigned and reimplemented to handle many different audio channels or different sampling rates. Another idea is, to convert the audio signal into the required audio formats of the watermarking scheme and cinema application. Then, this part can evaluate the watermarking scheme with the extended profile $P_{A-FormatConversion}$.

- **Transparency:** In a cinema, it is expected, that the quality of the audio content provides a very high quality level. This is needed to enjoy the listener (viewer of the movie) with excellent content. Therefore, the transparency of the used watermarking scheme must provide a very high transparency, which can be evaluated with the basic profile $P_{E/A/D-Transparency}$.

The results of the basic and extended profiles, which are part of the application profile Cinema, provide in its summary a recommendation for the evaluated watermark scheme in the application scenario of “Cinema”.
Perceptual Hash: This application profile [LDK07b] can be used to evaluate the embedding and attacking transparency of a digital audio watermarking scheme as well as the robustness of a perceptual hashing function.

In general, the evaluation within the application field of perceptual hashes can be considered from two completely different points of view. On one hand, one could focus on the evaluation of a digital watermarking scheme, whereby the perceptual hashing function is used to measure specific properties of the watermarking algorithm. On the other hand, properties of the perceptual hashing function can be evaluated with a digital watermarking scheme and its embedding and/or attacking function. Derived from both points of view and the motivation of enhancing the application profile evaluation, the application profile definition is introduced and formalized.

When evaluating a digital watermarking scheme with focus on an application scenario using perceptual hashes, the following questions arise:

1. How can an evaluation of given watermarking schemes be done providing comparability between the different schemes?
2. How can an assignment of the best fitting candidates for a given application scenario be achieved?
3. Does the embedding function of a digital watermarking scheme with the given embedding parameter set $p_E \in \mathcal{P}_E$ change the perceptual hash?
4. Does the attacking function with the given attacking parameter set $p_A \in \mathcal{P}_A$ change the perceptual hash?

Note that, derived from these questions, different evaluation methods for the different functions are required. Based on the profile based evaluation technique introduced in [LD04, LDLD05, LD06a], the focus is firstly set on the embedding function of a digital watermarking scheme and secondly on the attacking function by evaluating the transparency of both within the application field of perceptual hashing. Finally, the robustness of a perceptual hashing function can also be evaluated with digital audio watermarking schemes and thereby their impact on the computed perceptual hash.

**Evaluation of the Embedding Function with a Perceptual Hash**

The embedding of a message $m$ into the cover signal $S \in \mathcal{S}$ by using an embedding function $E$ of a digital watermarking scheme, results in signal modifications within the marked signal $S_E$. This can be the reason, that a perceptual hash, computed from the cover signal $S$, changes after embedding a digital watermark and the changes occurred by embedding need to be evaluated. Additionally, the selected embedding parameter set $p_E \in \mathcal{P}_E$ has an effect on the marked signal $S_E$ and therefore on its perceptual hash.

In this case, the evaluation of the embedding function $E$ is done within the application field of perceptual hashes. The goal is to measure the impact of the embedding function on the perceptual hash. The classification of this evaluation scenario is associated to the embedding profiles already introduced in [LDLD05] and defined as follows:

\[
P_{E-\text{Perceptual Hash}} = (\text{in–signal} \mid\mid \text{out–signal} \mid\mid \text{param})
\]

\[
\text{param} = (\text{alg} \mid\mid \text{alg-opt} \mid\mid \text{hashalg} \mid\mid \text{hashalg-opt})
\]
The parameter “in–signal” defines the input audio signal (in–signal = S ∈ S) and the parameter “out–signal” defines the marked output audio signal S_E. The parameter “alg” defines the watermarking scheme Ω, which is evaluated with its parameters defined with “alg-opt”. Note, that in this case “alg-opt” = p_E ∈ P_E. The used perceptual hashing function H is defined with “hashalg” and its required parameters with “hashalg-opt”.

The internals of this application profile are the measurement of the transparency T between S and S_E in the closed interval [0, 1] where 0 identifies the worst case (S and S_E are so different that S_E cannot be recognized as a modified version of S) and 1 is the best case (an observer does not perceive any significant difference between S_E and S).

\[ T(S, S_E) \rightarrow [0, 1] \quad (3.163) \]

The internals of the transparency measure function T(S, S_E) are in case of the application scenario of perceptual hashing, to identify, if both audio signals are equal or, if not, how close they are together. Therefore, the new transparency measure called Perceptual Hash Difference Grade (PHDG) is defined, formalized and introduced in the following. The goal of this embedding profile is to evaluate the effect of the transparency of the embedding function of a given digital watermarking scheme Ω on the perceptual hash shown in Figure 3.7. After selecting the cover signal S ∈ S and the embedding parameter set p_E ∈ P_E, the general principle is as follows:

1. Compute the perceptual hash H(S) value from S and store it in the database.
2. Embed the watermark w, which contains the encoded message m, into S with the embedding function E and its required parameters p_E. The result is the marked signal S_E.
3. Compute the perceptual hash value H(S_E) from S_E.
4. Compare H(S) with H(S_E) and if H(S) = H(S_E), then E and the selected parameters p_E do not effect the perceptual hash (relative perceptual hash difference grade phdg_{rel} = 1). Otherwise, if H(S) ≠ H(S_E), then E and/or p_E result in audio signal modifications, that change the perceptual hash and the distance between H(S) and H(S_E) should be computed. The relative perceptual hash difference grade for the embedding function (phdg_{Erel}) is defined as follows.

\[
\text{phdg}_{E_{\text{rel}}} (\Omega^*, S, H) = 1 - \frac{1}{|H|} \sum_{i=1}^{|H|} H_i(S) \oplus H_i(S_E) \quad (3.164)
\]

Whereby Ω* defines the digital audio watermarking scheme with its required parameter set, S is the original audio signal and H the used perceptual hashing function. The audio signal S_E is derived from Ω* and S by embedding a digital audio watermark into S. Furthermore, ⊕ depicts the exclusive OR between the binary representation of both perceptual hash values and the index i runs over all bits from the perceptual hash. The result is normalized with the closed interval [0, 1], whereby 0 is the worst case (all bits of H(S) and H(S_E) are different) and
1 the best case (all bits are equal). With this definition it is possible to measure the distance between two given perceptual hash values by counting the differences and computing the hamming distance [Ham50].

\[
\text{phdg}_{E} = \sum_{S \in S} \text{phdg}_{E\text{rel}}(\Omega^*, S, H).
\]

(3.165)

• Maximum embedding transparency:

\[
\text{phdg}_{E\text{max}}(\Omega^*) = \max_{S \in S} \left\{ \text{phdg}_{E\text{rel}}(\Omega^*, S, H) \right\}.
\]

(3.166)

• Minimum embedding transparency:

\[
\text{phdg}_{E\text{min}}(\Omega^*) = \min_{S \in S} \left\{ \text{phdg}_{E\text{rel}}(\Omega^*, S, H) \right\}.
\]

(3.167)

Note, that, depending on the design, implementation or configuration of the perceptual hashing function \(H^*\), the output could be different with a changed focus on the computation of the perceptual hash. For example, if the perceptual hashing algorithm splits the whole audio signal into smaller frames and returns the perceptual hash value for each frame, then a frame based perceptual hash value can be computed. Alternatively, the output of the perceptual hash could focus on different frequency bands of the given audio signal and therefore, a frequency based output could be computed. Derived from both examples, which describe two exemplarily selected possible outputs of the
perceptual hash, it is necessary to identify them by adding the specific information as superscript on the transparency notation. For example, an average embedding transparency computed with a frame based perceptual hash would be noted as $\text{phdg}_{\text{frame}}^{\text{E}_{\text{ave}}}$, the minimum and maximum embedding transparency as $\text{phdg}_{\text{frame}}^{\text{E}_{\text{min}}}$ and $\text{phdg}_{\text{frame}}^{\text{E}_{\text{max}}}$.

**Evaluation of the Attacking Function with a Perceptual Hash**

The evaluation of the attacking function $A$ of a digital watermarking scheme is like the evaluation of the embedding function introduced in the previous subsection. The difference is, that instead of $E$, the attacking function $A$ modifies the audio signal and produces the attacked audio signal $S_{E,A}$. Derived from this modification, the attacking transparency of $A$ is measured to identify the impact on the perceptual hash. The formal definition of this attacking profile is given as follows:

$$P_{A-\text{PerceptualHash}} = (\text{in–signal} \| \text{out–signal} \| \text{param})$$

(param = (alg \| alg-opt \| hashalg \| hashalg-opt)

The parameter “in–signal” defines the input audio signal and the parameter “out–signal” defines the marked output audio signal. The parameter “alg” defines the attacking function $A \in \mathcal{A}$, which is applied with its parameters $p_A \in \mathcal{P}_A$ defined as “alg-opt”. The applied perceptual hashing function is defined as “hashalg” and its required parameters with “hashalg-opt”.

The usage of this attacking profile in order to evaluate the transparency of the attacking function with the perceptual hash is visualized in Figure 3.8. After selecting the audio signal $S' \in \mathcal{S}$ (whereas in many cases $S' = S_E$ is set) and the attacking parameter set $p_A \in \mathcal{P}_A$, the general principle is as follows:

1. Compute the perceptual hash value $H(S')$ from $S'$ and store it in the database.
2. Attack $S'$ with the selected attack $A$ and its parameters $p_A$. The result is the attacked signal $S_{E,A}$.
3. Compute the perceptual hash $H(S_{E,A})$ value from $S_{E,A}$.
4. Compare $H(S')$ with $H(S_{E,A})$ and if $H(S') = H(S_{E,A})$, then $A$ and the selected parameters $p_A$ do not effect the perceptual hash. Otherwise, if $H(S') \neq H(S_A)$, then $A$ and/or $p_A$ result in such audio signal modifications, the perceptual hash changes and the distance between $H(S')$ and $H(S_{E,A})$ should be computed. The relative perceptual hash difference grade for the attacking function ($\text{phdg}_{A_{\text{rel}}}$) is defined as follows.

$$\text{phdg}_{A_{\text{rel}}} (\Omega^*, S', H) = 1 - \frac{1}{|H|} \sum_{i=1}^{|H|} H_i(S') \oplus H_i(S_{E,A})$$

Whereas $\Omega^*$ defines the digital audio watermarking scheme with its required parameter sets, $S'$ is the input audio signal and $H$ the used perceptual hashing function. The audio signal $S_{E,A}$ is derived from $\Omega$ and $S'$ by attacking $S'$ and modifying the signal. Furthermore, $\oplus$ depicts the exclusive OR between the binary representation of both perceptual hash values and the index $i$ runs over all bits from the perceptual hash. The
maximum is defined with the length of the perceptual hash value noted as $|H|$. Thereby, the best relative attacking transparency is measured with a value of zero, which means no differences between $H(S')$ and $H(S_A)$ are identified. The worst case would be, that all bits are changed with a return value of the relative attacking transparency measurement of the length of the perceptual hash ($|H|$). Thereby, it is possible to measure the distance between two given perceptual hash values by counting the differences and computing the hamming distance [Ham50].

As written the phdg$_E$ values derived from the embedding function it is also possible to measure the frequency band and frame based views of the perceptual hash for the attacking function. Thereby, the results would be phdg$_{frame}$, phdg$_{frame}^{ave}$, phdg$_{frame}^{min}$, phdg$_{frame}^{max}$, phdg$_{fb}^{ave}$, phdg$_{fb}^{min}$, and phdg$_{fb}^{max}$ for the frame based view and phdg$_{fb}$ for the frequency band view.

If this scenario is slightly modified, then it can be used to evaluate the robustness of a given perceptual hashing algorithm like, was done in [ÖSM04], against specific selected attacks or specific attack parameters. Thereby, different attacks are selected and the perceptual hashes $H(S')$ and $H(S_{EA})$ are computed and compared. If these hash values are equal, then the perceptual hashing function is robust against the specific attack $A$. Otherwise, if the attack parameters $p_A$ are slightly changed, then the threshold of secure perceptual hash and fragile perceptual hash can be identified.

**Evaluation of a Digital Watermarking Scheme with Perceptual Hashes and Vice Versa**

The evaluation of a digital watermarking algorithm in the application field of a perceptual hashing function can be done with the embedding and attack evaluation steps introduced in the pages above. Derived from these, two general evaluation goals exist.

- The first evaluation goal can be the evaluation of a digital watermarking scheme...
using perceptual hashing. The computed perceptual hash is used as new transparency measurement (perceptual hash difference grade) of the embedding and/or attacking function of the digital audio watermarking scheme.

- The second test goal can be the robustness evaluation of the perceptual hashing function by embedding a digital watermark or attacking the audio signal with different attacks using the attacking function of a digital audio watermarking scheme, which represent malicious or non-malicious signal modifications.

For both different evaluation goals, the following Figure 3.9 visualizes the application scenario. In general, the embedding function $E$ embeds the digital watermark with the given parameter set $p_E \in \mathcal{P}_E$ into the audio signals $S \in \mathcal{S}$. For the input and output audio signal, the corresponding perceptual hashes $h^1 = H(S)$ and $h^2 = H(S_E)$ are computed and stored in the database. After, for example distributing the marked signal $S_E$, different signal modifications can occur, simulated with the attack function $A$ and its required parameter set $p_A \in \mathcal{P}_A$. From the marked, attacked signal $S_{EA}$ a perceptual hash is also computed $h^3 = H(S_{EA})$ and stored in the database. The comparison of the perceptual hash values stored within the database is used to evaluate on one hand the transparency of the embedding and/or attacking function of the digital watermarking scheme, or, on the other hand, the robustness of the perceptual hashing function. The computed results are the evaluation test results and can be used for a recommendation of the watermarking scheme within the application of the selected perceptual hashing function with the used parameter sets and test sets. Finally, the detection/retrieval functions $D$ and $R$ try to detect and/or retrieve the watermark to verify the successful embedding of $m$ into the given signal $S \in \mathcal{S}$.

![Figure 3.9: Example of the Application Profile “Perceptual Hashing” $P_A$–PerceptualHash](image)

Depending on the used application scenario of perceptual hashes and the derived application goals, a digital watermarking scheme can be evaluated to get a recommendation. For the evaluation itself, the application profile $P_A$–PerceptualHash is used and defined as follows:
The parameter “in–signal” is the input audio signal \( S \in S \) and the parameter “out–signal” defines the resulting output audio signal \( S_{EA} \). The parameter “alg” defines the watermarking scheme, which is evaluated with its parameters defined with “alg-opt”. The used perceptual hashing function is defined with “hashalg” and its required parameters with “hashalg-opt”. To compute the hash values \( h_1, h_2 \) and \( h_3 \), always the same hash function “hashalg” with the same parameter set is used. The attack, to modify the marked signal, is defined within “att” and its required parameter set in “att-opt”.

Depending on the addressed application goals, the evaluation steps are introduced in the following listing. Firstly, the focus is set on the transparency evaluation of the embedding and attacking function of the digital watermarking scheme and secondly on the robustness of the perceptual hashing function:

- If the transparency of the embedding function \( E \) and/or attack function \( A \) is evaluated, then the general principle can be seen separately as introduced in the previous pages. The embedding transparency \( \text{phdg}_E \) and/or attacking transparency \( \text{phdg}_A \) as defined above are computed.

- If the robustness of the perceptual hashing function is evaluated, then the signal modification occurred by embedding a digital watermark or attacking a marked signal are seen as “robustness attack” against the perceptual hash. Thereby, the “attack” goal is to change the signal with, or without changing the perceptual hash depending on the application scenario. With the predefined transparency measurements \( \text{phdg}_E \) and \( \text{phdg}_A \) the robustness measurement is seen as follows: The perceptual hash is robust against a specific embedded watermark or attack and a particular audio signal \( S \), if the transparency measurement is equal to 1.0 (\( \text{phdg}_{E,\text{rel}} = 1.0 \) or \( \text{phdg}_{A,\text{rel}} = 1.0 \)). Otherwise, the perceptual hash is not robust against the embedded watermark with the selected embedding parameter set or the selected attack with its parameter set.

**PodCast:** The application profile PodCast simulates the podcast of audio signal distribution by streaming over the Internet. The general idea of podcasting is introduced by Ben Hammersley [Unl06], whereby a broadcasted audio signal for iPods\(^5\) is distributed. The server or the author of a podcast audio signal is often called a podcaster. The difference of this application scenario to the application scenario of “Broadcast”, introduced above, is that the podcasted audio signal is often encrypted to provide confidentiality as part of a DRM system. The evaluation of a watermarking scheme with this application scenario is defines as follows.

\[
P_{A-\text{PodCast}}(\text{in–signal} \ || \ \text{out–signal} \ || \ \text{param})
\]

The parameter “in–signal” defines the input audio signal, whereby the output audio signal defines the parameter “out–signal”. The parameter “param” defines the properties and characteristic of the podcast application used for the evaluation. The main

\(^5\)An iPod is a brand of portable media players designed and marketed by Apple Computer.
characteristics can be evaluated by different basic and application profiles, which are introduced in the following itemization.

- **Broadcast**: As briefly introduced, the podcast application is based on broadcasting of audio signals. Thereby, the evaluation of watermarking schemes can be partly reduced to the evaluation with the application profile $P_{A-Broadcast}$.

- **Annotation**: In many cases, the podcasted audio signal includes additional information like, for example, interpret or title of a song. If this annotation is done by embedding the additional information into the audio signal, then the evaluation results of the extended profile $P_{E-Annotation}$ is part of this application profile “PodCast”.

- **Security**: In addition, the podcasted audio signal provides security features. Such a feature is, for example, encryption to provide confidentiality to protect the bought audio content from illegal copies. The effects of the used security mechanisms by using digital watermarking algorithms can be evaluated with the basic profile $P_{E/A-Security}$.

It is noted again, that the defined application profiles above introduces exemplary selected application profiles and should motivate to design new application profiles needed for the watermark evaluation and to show the usage of application oriented evaluation of watermarking schemes.

The following Table 3.8 summarizes the introduced application profiles and lists the composition of required basic and/or extended profiles, which are part of the application profile. Thereby, it is shown, that application profiles can be composed of basic and/or extended profiles as well as of other application profiles. The definition of an application profile can be reduced to the definition of the profiles, which must be composed.

<table>
<thead>
<tr>
<th>Application Profile</th>
<th>Composed of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basic Profile</td>
</tr>
<tr>
<td>$P_{A-Archive}$</td>
<td>$P_{A-Robustness}$</td>
</tr>
<tr>
<td>$P_{A-Biometrics}$</td>
<td>$P_{E-Complexity}$</td>
</tr>
<tr>
<td>$P_{A-Broadcast}$</td>
<td>$P_{E-Complexity}$</td>
</tr>
<tr>
<td>$P_{A-Cinema}$</td>
<td>$P_{E-Complexity}$</td>
</tr>
<tr>
<td>$P_{A-PerceptualHash}$</td>
<td>$P_{E-Complexity}$</td>
</tr>
<tr>
<td>$P_{A-Podcast}$</td>
<td>$P_{E/A-Security}$</td>
</tr>
</tbody>
</table>

Table 3.8: Summarization of Application Profiles and the Composition of them
3.1.6 Evaluation Methodology

After defining and formalizing the watermarking parameters with their measurements in section 3.1.1 as well as derived from these formalizations the profile based evaluation in section 3.1.5, this section describes and summarizes the evaluation methodology. On one hand the watermark evaluation is based on the watermark parameters with their measurement and on the other hand on the profile based evaluation. Both strategies are summarized in the following sections 3.1.6.1 and 3.1.6.2, whereby their definitions and associations to the embedding, attacking and detection/retrieval functions are shown.

3.1.6.1 Evaluation Methodology based on Watermarking Parameters

From the introduced parameters and measure methods from subsection 3.1.1 an evaluation methodology can derived now to analyze one given watermarking algorithm (intra–algorithm evaluation or analysis) and to compare different watermarking algorithms (inter–algorithm evaluation or analysis). The evaluation methodology uses all defined parameters of a watermarking scheme and measures are summarized in Table 3.9. The idea is to describe firstly the general parameters for each watermarking algorithm and secondly the achieved results from the embedding, detection/retrieval and attacking functions for each algorithm itself as well as in comparison to other. If the algorithm itself is analyzed, it might be of interest to consider different parameter settings of embedding, detection and retrieval parameters and its influence to the watermarking properties as well as the specific behavior to a specific attack parameter setting on a selected test set. Furthermore in the case of a comparison of different algorithms it might be of interest to determine the best algorithm where the different measures allow to specify a certain objective to achieve (i.e. the overall transparency as average function or minimal transparency as lower bound).

<table>
<thead>
<tr>
<th>Object or Signal</th>
<th>Embedding Function</th>
<th>Detection/Retrieval Function</th>
<th>Attacking Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Omega(E, D, R, M, \bar{P}_E, \bar{P}_D, \bar{P}_R)$</td>
<td>$p_E \in \bar{P}<em>E$, $\text{cap}</em>{Erel}$, $\text{cap}_{Eave}$</td>
<td>$p_D \in \bar{P}<em>D$, $p_R \in \bar{P}<em>R$, $\text{cap}</em>{Rel}$, $\text{cap}</em>{Rrel}$, $\text{det}_{Rave}$</td>
<td>$p_A \in \bar{P}<em>A$, $\text{cap}</em>{Arel}$, $\text{cap}<em>{Aave}$, $\text{det}</em>{Aave}$</td>
</tr>
<tr>
<td>$m, m'$</td>
<td>$\text{cap}<em>{Rrel}$, $\text{cap}</em>{Rrel}$, $\text{det}_{Rave}$</td>
<td>$\text{cap}<em>{Arel}$, $\text{cap}</em>{Aave}$, $\text{det}_{Aave}$</td>
<td></td>
</tr>
<tr>
<td>$w$</td>
<td>$\text{det}<em>{D}$, $\text{det}</em>{Dave}$</td>
<td>$\text{det}<em>{A}$, $\text{det}</em>{Aave}$</td>
<td></td>
</tr>
<tr>
<td>$S, S_E$</td>
<td>$\text{tra}<em>{Erel}$, $\text{tra}</em>{Eave}$, $\text{tra}<em>{Emin}$, $\text{tra}</em>{Emax}$</td>
<td>$\text{det}_{A}$</td>
<td></td>
</tr>
</tbody>
</table>
Table 3.9: Summarizing of Evaluation Methodology based on Watermarking Parameters

The methodology therefore requires firstly the definition of all possible parameters needed by the embedding, detection/retrieval and attacking functions to setup $\Omega$ for a specific watermarking algorithm. These parameters are needed to compare different parameter settings or different test set classifications for one algorithm (intra-algorithm analysis) as well as to compare different parameter settings and test sets between different algorithms (inter-algorithm analysis) of all functions $E$, $D$, $R$ and $A$.

Secondly this methodology evaluates the algorithm with different input and output parameters, summarized in the first row by measuring the embedding, detection/retrieval and attacking performance with the measures summarized in the rows of the second, third and fourth columns. With this methodology an (one) algorithm can be tested with different parameter settings to compare the different performance results from these different parameter setting (intra-algorithm analysis). The tests, for example, can compare the influence of different attack parameter settings to one specific embedding and detection/retrieval setting to one algorithm. Furthermore, if different algorithms are compared, it is possible to place the algorithms in the same or in a different magic triangle (see page 11) depending on the test results in order to show the performance differences (inter-algorithm analysis).

In particular the evaluation of capacity for embedding or retrieval depends on $m$ and $m'$. For embedding, $\text{cap}_E^*$ defines the absolute length of $m$ and $\text{cap}_{E,rel}^*$ the relative length of $m$ normalized to the length of the audio signal. For retrieval, $\text{cap}_{R,rel}^*$ defines the absolute
lengths of retrieved $m'$. Therefore, it is used to measure for example the bit error rate (BER) or byte error rate (BYR) over the whole audio signal. A repeated embedding of $m$ can be identified as well in the $\text{cap}^\dagger_{Rrel}$. The retrieved capacity can be normalized to the length of the audio signal (or frames of it) with $\text{cap}_{Rrel}$ or to the length of $m$ with $\text{cap}^\dagger_{Rrel}$. For attacking, the capacity $\text{cap}_{Aave}$ defines the normalized average capacity after one or more attacks to an audio test set. Furthermore, $\text{cap}_{Amin}$ and $\text{cap}_{Amax}$ defines the minimum and maximum received capacity after one or more attacks on a given audio test set. The function $\det_{\tau}$ for a zero-bit watermarking scheme and $\det_{R}$ for $n$-bit watermarking scheme determines, if a given $m$ can be embedded into an audio signal or not. Therefore, the average values of $\det_{Dave}$ and $\det_{Rave}$ show the average success of the embedding function by using them directly after embedding the detection or retrieval function as verification.

The transparency of the embedding function (between $S, S_E$) can be measured with $\text{tra}_{Erel}$ for a specific watermarking algorithm and a specific audio signal with a given parameter set. Furthermore, $\text{tra}_{Eave}, \text{tra}_{Emin}$ and $\text{tra}_{Emax}$ defines the average, minimal and maximal transparency of a watermarking algorithm applied to a test set. The attacking transparency between the marked and attacked signal ($S_E, S_{EA}$) is similar measured to the embedding transparency. Therefore, the relative ($\text{tra}^*_{Arel}$), average ($\text{tra}_{Aave}$), minimal ($\text{tra}_{Amin}$) and maximal ($\text{tra}^*_{Amax}$) attacking transparency can be measured and compared. If the attacking transparency is measured between the attacked and original audio signal ($S, S_{EA}$), then the same types of transparencies are defined: relative ($\text{tra}^*_{Arel}$), average ($\text{tra}^*_{Aave}$), minimal ($\text{tra}^*_{Amin}$) and maximal ($\text{tra}^*_{Amax}$). The functions $\det_{D}$ and $\det_{R}$ measure the positive detection of $m'$. Therefore, the result is 0 (zero), if $m \neq m'$ and 1, if $m = m'$ at least once for a given audio signal. The average result over a test set is measured with $\det_{A}$, which is in the range of $[0, 1]$.

The robustness of a watermarking algorithm based on the bit or byte error rate can be measured with the average over the whole test set ($\text{rob}_{ave}^{\text{byte}}, \text{rob}_{ave}^{\text{bit}}$), the minimum ($\text{rob}_{min}^{\text{byte}}$) which includes the best attacking transparency and the best detection/retrieving results and a probabilistic result ($\text{rob}_{prob}^{\text{byte}}$). Therefore, $m'$ is retrieved with function $R$ of $\Omega^*$ and $m$ must be known to measure $\text{cap}^\dagger_{Rrel}$. For $\text{rob}_{ave}^{\text{bit}}$, the thresholds $\tau$ and $\nu$ define with the function $\det_{R\tau}$, if $\Omega^*$ is robust or not against $A_{i,j}$ by using a detection of $m'$ depending on $\tau$. Furthermore, the results of $\det_{A\tau}$ for a specific or $\det_{Aave}$ for all attacks depict the average of successful detection. If no threshold is needed, because the application scenario requires the complete message, then $\det_{A}$ and its average values are measurable. This result is a byte error rate because it is successfully only if at least once $w$ can be detected for zero-bit or $m' = m$ retrieved for $n$-bit watermarking schemes.

The complexity of a watermarking scheme is comparable, if the same message and the same audio signal is used for evaluating. Otherwise, the measured complexity must be normalized. Without the normalization, the relative complexity is measured for embedding, detection/retrieval and attacking with $\text{com}^*_{Erel}, \text{com}_{Drel}, \text{com}_{Rrel}$ and $\text{com}_{Arel}$. The defined measures allow the methodology to evaluate with two normalizations of the relative complexity. On one hand, the measured complexity result can be normalized by the length of the audio signal (size of $S, S_E$ or $S_{EA}$), which must be known, whereby the complexities for embedding, detection/retrieval and attacking are $\text{com}^S_{Erel}, \text{com}^S_{Drel}, \text{com}^S_{Arel}$ and $\text{com}_{Arel}$ measured. On the other hand, the normalization done by the length of the message (capacity)
provides the complexity measures for embedding and detection/retrieval $\text{com}^{C}_{E_{\text{rel}}}$, $\text{com}^{C}_{R_{\text{rel}}}$ and its comparability. For zero-bit watermarking schemes, this normalization cannot be done, as no message exists and the attacking function does not have access to the message, whereby it is not measurable.

The two other properties of watermarking schemes, invertible and verification can be seen as flags. If, for example, an inter–algorithm evaluation or analysis over a large set of watermarking schemes $\Omega_{1}^{*}, \ldots, \Omega_{n}^{*}$ is performed and the evaluation scenario requires only blind watermarking schemes ($\text{ver}(\Omega_{i}^{*}, S_{E}) = 1$, with $i = 1, \ldots, n$), then all watermarking schemes, where $\text{ver}(\Omega_{i}^{*}, S_{E}) \neq 1$ are masked out and cannot be used for the evaluation to achieve the requirements. The methodology uses the same method for the invertibility flag $\text{inv}$. If a specific requirement regarding the invertibility is given, then only the watermarking schemes which achieve the invertibility requirements are used for evaluation.

The introduced methodology based on watermarking properties allows intra– and inter–algorithm evaluation or analysis as well as the separate selection of embedding, detection/retrieval parameters for $\Omega^{*}$, the attacking functions and its parameters, the test set $S$ and the overall attack set $A$.

3.1.6.2 Evaluation Methodology Based on Profiles

From the introduced basic, extended and application profiles above, an evaluation methodology can be derived. With this methodology an intra– and inter–algorithm evaluation or analysis of one or many watermarking schemes can be done to provide comparability between given parameter and/or test sets or between different watermarking schemes. The evaluation uses the predefined profiles with their measurements for the embedding, attacking and detection/retrieval functions, which is summarized in Table 3.10. In addition, the profile based evaluation can be easily enhanced with other new defined and created profiles. The idea is, to describe the different views on the evaluation methodology. One view are users, researchers or algorithm designers, who, for example, want to know the detailed properties of an algorithm or they want to tune these properties. The other view can be done by users, which want to use or employ a watermarking scheme. The last view on this methodology is the view on the processes of the watermarking scheme. Thereby, the users can be interested in the properties of embedding, detection/retrieval and/or attacking function on different views.
The methodology therefore requires, like by the evaluation methodology based on watermarking parameters (section 3.1.6.1), all possible parameters for the embedding and detection/retrieval for a watermarking scheme. Furthermore, all possible parameter sets of the attacking functions, needed by the profiles, must be defined. With both, an intra-algorithm evaluation and analysis of a given watermarking scheme or an inter-algorithm evaluation and analysis of many given watermarking schemes, based on a profile based evaluation, can be done.

The three categories of profiles (basic, extended and application) are designed to provide different views of the evaluation to the user. Thereby, the single properties of a watermarking scheme, which are mostly interesting for watermark algorithm designers and developers, can be evaluated with the basic profiles. Thereby, specific properties and parameters are given and required, which can be for a non-expert difficult to understand. In contrast, if somebody wants to use a digital watermark in an application scenario, for example in order to annotate the signal with additional information, then she/he is not interested in detailed technical information about the watermark properties and internals. She/he wants to know, which watermarking scheme can be recommended for the specific application scenario and which not. To provide such recommendations, the application profile based evaluation was designed.

If a new upcoming watermarking algorithm is evaluated, then it is mostly benchmarked with a large test of digital medias used for embedding, attacking and detection/retrieval. Therefore, the test results must be comparable to each other and existing benchmarking results as well as it is often necessary, to compute average values of the test results. There are different possibilities available. One and typical one is the average function, which computes

<table>
<thead>
<tr>
<th>Basic Profiles</th>
<th>Embedded Function</th>
<th>Detection/Retrieval Function</th>
<th>Attacking Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE−Transparency</td>
<td>PD−Complexity</td>
<td>PA−Capacity</td>
<td></td>
</tr>
<tr>
<td>PE−Capacity</td>
<td>PD−Complexity</td>
<td>PA−Complexity</td>
<td></td>
</tr>
<tr>
<td>PE−Complexity</td>
<td>PD−Complexity</td>
<td>PA−Robustness</td>
<td></td>
</tr>
<tr>
<td>PE−Invertibility</td>
<td>PA−Security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PE−Security</td>
<td>PA−Security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PE−Verification</td>
<td>PA−Transparency</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Extended Profiles</th>
<th>Embedded Function</th>
<th>Detection/Retrieval Function</th>
<th>Attacking Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE−Annotation</td>
<td>PD−Calculation_Time</td>
<td>PA−Calculation_Time</td>
<td></td>
</tr>
<tr>
<td>PE−Calculation_Time</td>
<td>PD−Combined_Algorithm</td>
<td>PA−DA/AD</td>
<td></td>
</tr>
<tr>
<td>PE−Combined_Algorithm</td>
<td>PD−Long_Time</td>
<td>PA−Estimation_Attacks</td>
<td></td>
</tr>
<tr>
<td>PE−Long_Time</td>
<td>PA−Format_Conversion</td>
<td>PA−Key_Space</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PA−Long_Time</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PA−Lossy_Compression</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PA−Packet_Loss</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.10: Summarizing of Evaluation Methodology based on Profiles
the average of a set of test results. The following equation introduces it in detail.

\[ \bar{x} = \frac{1}{|S|} \sum_{i=1}^{|S|} x_i \]  

(3.174)

Where \(|S|\) is the number of used audio files for the evaluation and \(x_i\) with \(i \in \mathbb{N}\) the evaluation result of the \(i\)th element and the specific evaluation function. The computed mean value can be used to compute the root mean square deviation introduced in the following equation.

\[ \sigma_x = \sqrt{\frac{1}{|S| - 1} \sum_{i=1}^{|S|} (x_i - \bar{x})^2} \]  

(3.175)

Therefore, the root mean square deviation (standard deviation) of \(\bar{x}\) can be computed as:

\[ \sigma_{\bar{x}} = \frac{\sigma_S}{\sqrt{|S|}} = \sqrt{\frac{1}{|S|}(|S| - 1) \sum_{i=1}^{|S|} (x_i - \bar{x})^2} \]  

(3.176)

In the field of benchmarking, the predefined variables \(x, x_i\) and \(|S|\) must be filled out with the profiles used for evaluation. Therefore, \(|S|\) is always the number of audio files used for the evaluation. In this work, only the average value of measured evaluation results is used, but it is shown, that the evaluation methodology is open for other measurements.

### 3.1.7 Audio Data Test Set: Formalization and Example Test Sets

The audio test set used for the digital audio watermarking algorithms evaluation and benchmarking is very important and has an impact on the evaluation results. Furthermore, the idea of the audio content dependency of the evaluated digital audio watermarking schemes is part of the profile based evaluation approach. Therefore, in this section the importance of the selected audio test set (in this work noted as \(S\)), used for digital audio watermark evaluation and the characteristic of the later used audio signals are described, which needs to satisfy an application oriented evaluation methodology. Derived from the watermark evaluation, which is content sensitive, the idea is to define the audio test set with many different types of audio signals (also called audio content) classified into different audio contents.

For the evaluation and benchmarking of digital audio watermarking schemes, the used test set, its characteristics and the amount of data is very important, because the evaluation results depends extremely on it [LHD04]. The evaluation results can only be compared to each other, if the same audio test set is used for the evaluation. By evaluating the properties of a given watermarking scheme with one or more basic profiles, defined in our methodology, it is recommended to use an audio test set, which includes most as possible audio files with different characteristics and types of content to simulate all possible application scenarios [LD06a]. Note, that with increasing number of audio test files, the complexity of all evaluation tests will increase too. If a watermarking scheme should be evaluated for a specific application scenario, with the goal of an application oriented benchmarking, then the audio test set should be include exactly these audio files, which are used by this application. If, for example, an
application scenario works only with speech audio signals, then the evaluation should be done with different speech signals.

The audio test set, which is defined and selected in this report, consists of 389 different and exemplary selected audio files, which are divided into the four main categories identified below containing royalty-free and licensed audio signals. The general idea is, to split all audio signals into the main categories music, speech, sounds and sqam and each of the main class into sub classes or sub categories which include about 20 audio signals per sub category. The four main categories for the audio signals and its classification into their sub-categories are as follows, but note that of course other audio categories with more or less number of audio signals are possible and these classifications are selected to show and demonstrate the exemplary selected audio test set for the profile based watermark evaluation approach.

**Music:** includes totally 265 audio files, which are distributed to ten sub-categories: metal, pop, reggae, blues, jazz, techno, hip hop, country, classical and synthetic. Each of these sub-category contains 20 exemplary selected audio signals. Additionally, the sub-category classical, with an additional 85 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 19 and opera 19 audio files.

**Speech:** includes totally 75 audio files, which are distributed to four sub-categories: male, female, computer generated and sports. These sub-categories contains male 24, for female 20, for computer generated 20 for sports 11 audio files.

**Sounds:** includes totally 33 audio files, which are distributed into four sub-categories: computer generated, natural, silence and noise. In computer generated are 12, natural 8, silence 2 and noise 11 audio files.

**Sqam:** includes 16 audio files, which is the well known SQAM test set [SQA] used extensively for testing. The sub-categories are instrumental with 7, speech with 6 and singing voice with 3 associated audio files. This test set is often used for smaller tests and as subset additionally introduced in Table 3.11.

The following Figure 3.10 visualizes the audio test set, which is defined as target in the enhanced CERT taxonomy [Lan08]. Note, the defined classification is an example and other classifications are possible. In the center of this figure is a rectangle “audio signals” presented. From there are the four main categories and their associated sub-categories drew.
To provide an adequate audio quality, the same audio characteristics as used for audio CD are used for the chosen audio test set. Therefore, all audio files are pulse code modulated (PCM) coded WAV\textregistered\ files with 44100 Hz sampling rate ($f_{SR} = 44.1$ kHz), 16 bit quantization ($Q = 16$ bit) and 2 channels (stereo), equating to standard audio CD format. The complete play time of all audio files has an amount of 3 hours, 5 minutes and 4.7 seconds. Therefore, the average play time is 28.55 seconds and the standard deviation of the playtime is 8.95. The shortest audio file, with 1.96 seconds, is a phone number, categorized into the main category sounds and sub-category noise. In contrast, the biggest audio file has a play time of 2 minutes and 4.8 seconds, contains spoken female speech and is categorized into the main class speech and sub-category female.

Note, the defined audio test set $S$ is open on one hand, to increase the number of audio signals and on the other hand the granularity of the main and sub categories for a more detailed audio content evaluation and analysis.

If it is known, that the watermark evaluation is time consuming or, if only a general tendency of the evaluation results is determined, then a subset of the complete audio test set $S$ can be chosen and used for the evaluation. The group $sqam$, for example, from the four main categories with the 16 SQAM files [Wat88] are selected (with $S_{SQAM} \subset S$) which contains speech, singing and instrumental audio signals. Note, this audio test set is used often in the literature for digital audio watermark evaluation, but in this report, it is classified into three sub categories and used as subset of the complete audio test set. The minimal length of an audio signal of $S_{SQAM}$ is 16.3s, the maximum length 34.9s and the average length of all 16 audio signals 21.26s. Furthermore, the audio files are categorized in three types of content,
which is shown in Table 3.11. Therefore, the first category \textit{single instrument} contains 7 audio files, where a single music instrument is audible, the second category \textit{speech} contains spoken text with female and male voices in the languages English, German and French. The last category \textit{singing} contains female, male and a mixture of both singing voices.

<table>
<thead>
<tr>
<th>Single Instrument</th>
<th>Speech</th>
<th>Singing</th>
</tr>
</thead>
<tbody>
<tr>
<td>harp40_1.wav</td>
<td>spfe49_1.wav</td>
<td>bass47_1.wav</td>
</tr>
<tr>
<td>horn23_2.wav</td>
<td>spfl51_1.wav</td>
<td>sopr44_1.wav</td>
</tr>
<tr>
<td>trpt21_2.wav</td>
<td>spfl53_1.wav</td>
<td>quar48_1.wav</td>
</tr>
<tr>
<td>vioo10_2.wav</td>
<td>spmc50_1.wav</td>
<td></td>
</tr>
<tr>
<td>gspi35_1.wav</td>
<td>spmc52_1.wav</td>
<td></td>
</tr>
<tr>
<td>gspi35_2.wav</td>
<td>spmg54_1.wav</td>
<td></td>
</tr>
<tr>
<td>frer07_1.wav</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.11: SQAM Audio Files ($S_{SQAM} \subset S$) and their Classification

The following Figure 3.11 visualizes the distribution of main and sub-categories of the audio test sets $S$ and $S_{SQAM}$. Thereby, it is shown, that the category “Music” contains the most and “SQAM” the fewest audio files. The sub-category “Classical” contains a similar number of audio files like the main category “Speech”.

![Distribution of Audio Signals](image)

Figure 3.11: Audio File Distribution of the Audio Test Set $S$ and its Main Categories
3.2 Practical Framework

Based on the theoretical framework, introduced in section 3.1, the practical framework is presented in this section to demonstrate the usability of the invented theoretical framework and the applicability of application oriented benchmarking for different profiles whereby the audio content dependency can be identified. The test goals for the application profile oriented evaluation on the example of perceptual hashing are described in order to provide an orientation for the test setup. The description of the test goals also refers to the later described test scenarios including the test sets. In this context also the selected digital audio watermarking schemes for the evaluation are introduced in subsection 3.2.2 in order to present the used parameter sets for the embedding and detection/retrieval functions of the watermarking schemes. Finally, the test scenarios with the used test setups are closing this chapter in subsection 3.2.3.

3.2.1 Test Goals

In this section, the test goals of the practical evaluation are defined and introduced.

The test goal $\Pi_1$ focuses on the application scenario of perceptual hashing, whereby the two test goals $\Pi_1$ and $\Pi_2$ are defined. The embedding transparency of the selected watermarking schemes is measured as average, minimum and maximum embedding transparency with the embedding profile $P_{E-Perceptual\,Hash}$ for both test goals. Thereby, test goal $\Pi_1$ evaluates $\text{phdg}_{E_{\text{ave}}}^{\text{fb}}$, $\text{phdg}_{E_{\text{min}}}^{\text{fb}}$ and $\text{phdg}_{E_{\text{max}}}^{\text{fb}}$, the average, minimum and maximum difference (distance) between the perceptual hash of the original ($S$) and marked audio signal ($S_E$) in the frequency band view.

Whereby the test goal $\Pi_2$ analyzes the same test results, but focuses on the time based view and computes $\text{phdg}_{E_{\text{ave}}}^{\text{frame}}$, $\text{phdg}_{E_{\text{min}}}^{\text{frame}}$ and $\text{phdg}_{E_{\text{max}}}^{\text{frame}}$. For test goal $\Pi_1$ a frequency depending watermark embedding and for test goal $\Pi_2$ a time depending watermark embedding of the selected watermarking schemes should be able to be identified. For both test goals $\Pi_1$ and $\Pi_2$ an intra- and inter-algorithm evaluation and analysis of the test results is presented.

The following table 3.12 summarized the two defined test goals.

<table>
<thead>
<tr>
<th>Test Goal</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Pi_1$</td>
<td>Evaluates the embedding transparency of the selected watermarking schemes with the embedding profile $P_{A-Perceptual,Hash}$, whereby the frequency band view is chosen ($\text{phdg}<em>{E</em>{\text{ave}}}^{\text{fb}}$).</td>
</tr>
<tr>
<td>$\Pi_2$</td>
<td>Evaluates the embedding transparency of the selected watermarking schemes with the embedding profile $P_{A-Perceptual,Hash}$, whereby the time based view is chosen ($\text{phdg}<em>{E</em>{\text{ave}}}^{\text{frame}}$).</td>
</tr>
</tbody>
</table>

Table 3.12: Summarization of the two defined Test Goals
3.2.2 Selected Watermarking Schemes for Evaluation

In this subsection, the exemplary selected digital audio watermarking schemes used for the application profile based evaluation, their formal description and their parameter sets of the embedding and detection/retrieval functions are introduced.

For the exemplary evaluation six different audio watermarking algorithms ($\Omega_\ast$)\textsuperscript{6} are selected with the focus on two implementations for algorithms working in time-, frequency- and wavelet domain. The following description summarizes briefly the selected digital watermarking schemes, contains the general parameter description and some more internals by describing the working domain of the functions $E$, $D$ and $R$ as additional information for a classification of the test results. For the later test setup the watermarking algorithms are seen as black boxes.

$\Omega_{2A2W}$: This semi-blind watermarking algorithm, is a $n$-bit watermarking algorithm. It embeds $m$ once, works in the wavelet domain and embeds the watermark on selected zero tree nodes [SCP93]. It does not use a secret key and can therefore categorized, from the application point of view, as an annotation watermarking scheme. An additional file is created, where the marking positions are stored to retrieve the watermark information in detection/retrieval function (non blind) [IMYK98]. By using $\Omega_{2A2W}$, the following parameters are defined for this algorithm:

- $2A2WE$: specifies the internal embedding method and at present only $ZT$ (zerotree) is implemented.
- $2A2WC$: specifies the internal coding method and at present, only binary (BIN) is possible. As $p_{\text{cod}} \in p_E$ the coding method used for $p_{\text{cod}}$ is seen as $p_{E}$.

**Embedding Function:** As input audio signal $S$, this watermarking scheme reads only uncompressed PCM audio files in W AVE format. The output signal $S_E$ is only writable in uncompressed PCM WAVE file format. The parameters needed for $E$ are $p_E = (2A2WE, 2A2WC)$.

**Detection/Retrieval Function:** As input audio signal $S_E$ or $S_{EA}$ only uncompressed PCM audio files in WAVE format are supported. Furthermore, there is no distinction between the detection and retrieval function ($D$ and $R$). Therefore, only the retrieval function $R$ can be used. The parameters needed for $R$ are $p_R = (2A2WE, 2A2WC)$, $D = \emptyset$.

The introduced parameters are subsequently assigned to $p_E$, $p_D$ and $p_R$. Therefore, this watermarking scheme can be described as follows:

$$\Omega_{2A2W} = (E, \emptyset, R, m, \{2A2WC = BIN, 2A2WE = ZT\}, \emptyset, \{2A2WC = BIN, 2A2WE = ZT\})$$

\textsuperscript{6}Whereby the star $\ast$, indicates a specific selected watermark scheme.
Other parameter combinations are currently not available. The working domain of this algorithm is wavelet and can exemplary be described as:

\[ \Omega_{2A2W} = (E_{\text{wavelet}}, \emptyset, R_{\text{wavelet}}, m, \{2A2W_C = \text{BIN}, 2A2W_E = \text{ZT}\}, \emptyset, \{2A2W_C = \text{BIN}, 2A2W_E = \text{ZT}\}) \]

(3.178)

\[ \Omega_{\text{LSB}}: \] This blind watermarking algorithm, embeds the watermark message \( m \) into the Least Significant Bits (LSB) of the audio signal. Thereby, this watermarking algorithm has two general modes [KL05], with and without the usage of a 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. The key initializes a PRNG and the values of the PRNG scramble the embedding position. This means, that not all LSBs are used and it is expected, that the capacity decreases and the transparency increases. Furthermore, it is assumed, that an enabled error correction code decreases the capacity and increase the robustness against randomly signal distortions. The implementation of the LSB watermarking scheme has the following parameters:

- \( \text{LSB}_k \): secret key to initialize the PRNG which is used for the scrambling mode
- \( \text{LSB}_c \): flag for ECC, is only ON or OFF, \( \{0, 1\} \)
- \( \text{LSB}_t \): defines the sample layout mode, selected from \( \{1, 2, 3, 4, 5\} \) to define the handling of more than one audio channel. Default is \( \text{LSB}_c = 0 \)
- \( \text{LSB}_x \): flag, if the synchronization pattern are fixed or not \( \{0, 1\} \). If this flag is 0, then exactly the same synchronization pattern between the multiple embedded \( m \)'s is used. Otherwise, it changes permanently, which increases the detection/retrieval complexity. Default is \( \text{LSB}_c = 0 \)
- \( \text{LSB}_j \): integer value, which defines the maximum jumping length in scramble mode. It means, if the secret key \( k \) is used, then the maximal scrambling length (not usage of all audio sample values) is defined with this parameter. It must be: \( \text{LSB}_j > 0 \) and \( \text{LSB}_j \in \mathbb{N} \). Default if \( \text{LSB}_x = 9 \).
- \( \text{LSB}_u \): integer value, which defines the maximum number of retries, to find the correct synchronization in case of use \( \text{LSB}_z \). Default is \( \text{LSB}_u = 5 \).

**Embedding Function:** As input audio signal \( S \), all audio file formats provided by the libsndfile library [dCL] are supported. But the focus is set on the uncompressed PCM audio WAVE format. The output signal \( S_E \) can also be written in all audio files formats provided by the library. Derived from the parameters, the embedding parameter set require the parameters \( p_E = \{\text{LSB}_k, \text{LSB}_c, \text{LSB}_t, \text{LSB}_x, \text{LSB}_j\} \) for the embedding function. If a parameter is missing, then the default value is used.

**Detection/Retrieval Function:** As input audio signal \( S_E \) or \( S_{EA} \), all audio file formats provided by the libsndfile library [dCL] are supported. But the focus is set on the uncompressed PCM audio WAVE format. There is no distinction between \( D \) and \( R \) possible. Therefore, only the retrieval function \( R \) can be used. Required parameters for \( R \) are \( p_R = \{\text{LSB}_k, \text{LSB}_c, \text{LSB}_t, \text{LSB}_j, \text{LSB}_u\} \).

Therefore, the parameter sets for \( p_E \) and \( p_R \) can be used for \( \Omega_{\text{LSB}} \), whereby \( p_D \) is empty.
This watermarking algorithm can be described as follows:

\[ \Omega_{\text{LSB}} = (E, \emptyset, R, m, \{\text{LSB}_k, \text{LSB}_c, \text{LSB}_t, \text{LSB}_x, \text{LSB}_j\}, \emptyset, \{\text{LSB}_k, \text{LSB}_c, \text{LSB}_t, \text{LSB}_j, \text{LSB}_u\}) \]  

The working domain of this algorithm is time domain and therefore, it can also be described with:

\[ \Omega_{\text{LSB}} = (E_{\text{time}}, \emptyset, R_{\text{time}}, m, \{\text{LSB}_k, \text{LSB}_c, \text{LSB}_t, \text{LSB}_x, \text{LSB}_j\}, \emptyset, \{\text{LSB}_k, \text{LSB}_c, \text{LSB}_t, \text{LSB}_j, \text{LSB}_u\}) \]

\(\Omega_{\text{PM}}\): This \(n\)-bit watermarking algorithm, embeds the message \(m\) into the cover signal by using an asymmetrical key pair for security reason. For the evaluation, only the compiled executable binary file is used and the available source code is not considered. The watermark\(^7\) embedding and detection/retrieval functions require only the key for security reason and therefore, it can be classified as copyright watermark.

**Embedding Function:** As input audio signal \(S\), the well known uncompressed PCM audio \texttt{WAV} format is supported (other formats provided by the library \texttt{audiofile} \[Pru\] should be supported too). The output signal \(S_E\) can also be written in uncompressed PCM \texttt{WAV} file format and all supported formats provided by the library \texttt{audiofile}. There is only the public key as parameter required for \(E\) defined.

- \(\text{PM}_{k_{\text{pub}}}\): specifies the public key used for embedding.

**Detection/Retrieval Function:** As input audio signals \(S_E\) or \(S_{EA}\) it is uncompressed PCM audio files in \texttt{WAV} format are supported (other formats provided by the library \texttt{audiofile} \[Pru\] should be supported too). Furthermore, there is also no distinguish between \(D\) and \(R\) possible. Therefore, only the retrieval function \(R\) can be used. The parameter for \(R\) is only the private key.

- \(\text{PM}_{k_{\text{priv}}}\): specifies the private key used for retrieval.

Therefore, only the parameter sets for \(p_E\) and \(p_R\) are fixed for \(\Omega_{\text{PM}}\), whereby \(p_D\) is empty. For an intra-algorithm analysis, only the test set, attack set and/or attacking parameters can be changed.

This watermarking algorithm can be described as follows:

\[ \Omega_{\text{PM}} = (E, \emptyset, R, m, \text{PM}_{k_{\text{pub}}}, \emptyset, \text{PM}_{k_{\text{priv}}}) \]  

The working domain of this algorithm seems to be time domain and therefore, it can also be described with:

\[ \Omega_{\text{PM}} = (E_{\text{time}}, \emptyset, R_{\text{time}}, m, \text{PM}_{k_{\text{pub}}}, \emptyset, \text{PM}_{k_{\text{priv}}}) \]

\(\Omega_{\text{MS}}\): This blind \(n\)-bit stream watermarking algorithm, works in the frequency domain and embeds the watermark in the frequency coefficients by using a spread spectrum technique [KM03]. It does not use a secret key and can therefore also be categorized as annotation watermarking scheme. This algorithm does not require a parameter for embedding and detection/retrieval.

\(^7\)It is also classified as steganographic scheme, but in this document, it is seen as watermarking scheme.
**Embedding Function:** As input audio signal $S$, the well known uncompressed PCM audio WAVE format is supported (more formats information are not available currently). The output signal $S_E$ can also be written in uncompressed PCM WAVE file format. There are no parameters for $E$ defined; ($p_E = (\emptyset)$).

**Detection/Retrieval Function:** As input audio signals $S_E$ or $S_{EA}$ it is uncompressed PCM audio files in WAVE format are supported (more formats are not known yet). Furthermore, there is also no distinguish between $D$ and $R$. Therefore, only the retrieval function $R$ can be used. The parameters required for $R$ are $p_R = (\emptyset)$.

Therefore, $\Omega_{MS}$ has no parameters for $p_E, p_D$ and $p_R$, which can be changed for different embedding or detection/retrieval parameter sets. For an intra–algorithm analysis, only the test set, attack set and/or attacking parameters can be changed.

This watermarking algorithm can be described as follows:

$$\Omega_{MS} = (E, \emptyset, R, m, \emptyset, \emptyset, \emptyset)$$ (3.183)

The working domain of this algorithm is the frequency domain and can be described with:

$$\Omega_{MS} = (E_{freq}, \emptyset, R_{freq}, m, \emptyset, \emptyset, \emptyset)$$ (3.184)

$\Omega_{SS}$: This blind $n$-bit stream watermarking algorithm, works in the frequency domain and embeds $w$ ($w = cod(m, p_{cod})$) in a selected frequency band by using a spread spectrum technique multiple times. Therefore a scaled sequence of random values is added to the frequency coefficients of the audio signal. This algorithm has the following parameters:

- $SS_k$: defines the secret key and is an integer value
- $SS_\alpha$: is the scaling factor used to define the embedding strength
- $SS_l$: defines the lower frequency bound in range $[0, f_{SR}/2]$
- $SS_h$: defines the upper frequency bound in range $[0, f_{SR}/2]$ and $SS_l \leq SS_h$
- $SS_f$: defines the frame size used for the windowing function typical power of 2
- $SS_\tau$: defines a threshold needed to retrieve $m'$ in range $[0, 1]$.

**Embedding Function:** As input audio signal $S$, this watermarking scheme is able to read and write all file formats provided by the `libsndfile` library [dCL]. The parameters needed for $E$ are $p_E = (SS_k, SS_\alpha, SS_l, SS_h, SS_f)$.

**Detection/Retrieval Function:** Supported input audio signals $S_E$ or $S_{EA}$ are all file formats provided by the `libsndfile` library. The implementation of $\Omega_{SS}$ does not distinguish between $D$ and $R$. Therefore, only the retrieval function $R$ can be used. The parameters needed for $R$ are $p_R = (SS_k, SS_l, SS_h, SS_f, SS_\tau)$.

The maximum frequency of the frequency bound depends on the sampling rate $f_{SR}$ and is defines as $f_{tot} = \frac{f_{SR}}{2}$ [Jer77]. $\Omega_{SS}$ can be described as follows:

$$\Omega_{SS} = (E, \emptyset, R, m, \{SS_k, SS_\alpha \in [0, \infty], SS_l \in [0, f_{tot}], SS_h \in [0, f_{tot}] \land SS_h \geq SS_l, SS_f = 2^x, x \in \mathbb{N}, \emptyset, \{SS_l \in [0, f_{tot}], SS_h \in [0, f_{tot}], t \in [0, 1], SS_f, SS_\tau\})$$ (3.185)
The constrain \( SS_l \leq SS_h \) needs to be satisfied. The working domain of this algorithm is also the frequency domain and can be described as:

\[
\Omega_{SS} = (E_{freq}, \emptyset, R_{freq}, m, SS_{\alpha} \in [0, \infty], \{SS_l \in [0, f_{tot}], SS_h \in [0, f_{tot}], SS_f = 2^x, x \in \mathbb{N}\}, \emptyset, \{SS_l \in [0, f_{tot}], SS_h \in [0, f_{tot}], SS_f, SS_{r} \}) \quad (3.186)
\]

\( \Omega_{VAWW} \): This watermarking algorithm, can be classified as a zero-bit watermark. It works in the wavelet domain and embeds the watermark in selected coefficients [DRA98]. To embed the watermark into the audio signal a three level DWT domain and a Daubechies 8-tap filter is used [DRA98]. The following parameters can be defined:

- \( VAWW_k \): defines the secret key as integer value
- \( VAWW_\tau \): defines a threshold, which selects the coefficients for embedding. The default value is \( VAWW_\tau = 40 \)
- \( VAWW_\alpha \): defines a scale factor and which describes the embedding strength. The default value is \( VAWW_\alpha = 0.2 \).

**Embedding Function:** As input audio signal \( S \), this watermarking scheme reads and writes all file formats provided by the \texttt{libsndfile} library [dCL]. The parameters needed for \( E \) are \( p_E = (VAWW_k, VAWW_\tau, VAWW_\alpha) \).

**Detection/Retrieval Function:** Supported input audio signals \( S_E \) or \( S_{EA} \) are all file formats provided by the \texttt{libsndfile} library. Only the detection is possible and the parameters for \( D \) are \( p_D = (VAWW_k, VAWW_\tau, VAWW_\alpha) \).

Therefore, \( \Omega_{VAWW} \) is a zero-bit watermarking scheme, only \( D \) can be used for detection. This watermarking algorithm can be described as follows:

\[
\Omega_{VAWW} = (E, D, \emptyset, \emptyset, p_E(VAWW_k, VAWW_\tau, VAWW_\alpha), p_D(VAWW_k, VAWW_\tau, VAWW_\alpha, \emptyset)) \quad (3.187)
\]

The working domain of this algorithm is the wavelet domain and can be described as:

\[
\Omega_{VAWW} = (E_{wavelet}, D_{wavelet}, \emptyset, \emptyset, p_E(VAWW_k, VAWW_\tau, VAWW_\alpha), p_D(VAWW_k, VAWW_\tau, VAWW_\alpha)) \quad (3.188)
\]

In the following, the properties of the six exemplary selected digital audio watermarking schemes are summarized. Thereby, the working domain, the requirement of a secret key, a possible multiple embedding, the class of \( n \) or zero-bit watermarking scheme and the number of changeable parameters are in Table 3.13 summarized. Note, a “n.a.” in the column “Multiple Embedding” means, that it is unknown, if the watermarking message is embedded multiple times (black box testing).
<table>
<thead>
<tr>
<th>Watermarking Scheme</th>
<th>Class</th>
<th>Working Domain</th>
<th>Key Require</th>
<th>Changeable Parameters</th>
<th>Multiple Embedding</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Omega_{2A2W}$</td>
<td>n-bit</td>
<td>wavlet</td>
<td>no</td>
<td>0</td>
<td>no</td>
</tr>
<tr>
<td>$\Omega_{LSB}$</td>
<td>n-bit</td>
<td>time</td>
<td>yes/no</td>
<td>5</td>
<td>yes</td>
</tr>
<tr>
<td>$\Omega_{MS}$</td>
<td>n-bit</td>
<td>frequency</td>
<td>no</td>
<td>0</td>
<td>n.a.</td>
</tr>
<tr>
<td>$\Omega_{PM}$</td>
<td>n-bit</td>
<td>time</td>
<td>yes</td>
<td>0</td>
<td>n.a.</td>
</tr>
<tr>
<td>$\Omega_{SS}$</td>
<td>n-bit</td>
<td>frequency</td>
<td>yes</td>
<td>7</td>
<td>yes</td>
</tr>
<tr>
<td>$\Omega_{VAWW}$</td>
<td>zero-bit</td>
<td>wavelet</td>
<td>no</td>
<td>2</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

Table 3.13: Summarized Classes of Evaluated Watermarking Schemes

The shown number of changeable parameters of the selected digital audio watermarking schemes $\Omega_*$ is important, if an intra–algorithm evaluation analysis should be done. If no different parameter sets exists, then the audio content dependency and the intra–algorithm evaluation can be analyzed.

### 3.2.3 Test Scenario

In this subsection the test scenario is described. Different evaluation strategies are used to introduce the practical usage of the theoretical framework and the profile based evaluation on example of perceptual hashing.

The introduced test goals from Table 3.12 are used to evaluate and objectively compare the selected digital audio watermarking algorithms with inter- and intra–algorithm evaluation and analysis and to show the usage of the practical framework on the example of perceptual hashing. Thereby, the theoretical framework is prototypically implemented to show on a practical example how to measure and compare the transparency of $E$ and $A$. Furthermore, the detectability of the watermark $w$ and/or the retrieveability of the message $m'$ in $S_E$ and $S_{EA}$ are measured after embedding and attacking. The relationship between attacking transparency and robustness is used to identify the successful attacks, as well as the relationship between robustness and capacity to show the effect of an attack. Therefore, the following subsections show the test scenarios together with the measured parameters to obtain the test goals.

For all evaluation tests, a specific and always the same hardware and operating system (OS) are used. This is important, because the complexity measurement focuses on the computation time of the CPU and if, for example, the CPU speed changes, then the computation time of the CPU changes too. The AMSL research group at Otto-von-Guericke University of Magdeburg, where the research for this work is done, has as the following hardware and OS for off-line audio watermark evaluation.

- CPU: 2 x Intel Xeon™, Hyper Threading (HT), 3.60 GHz, 2.0 MB cache size
- RAM: 8.0 GB, (giga byte)
- Haddisk: 1.4 TB (tera byte), RAID level 5
• Swap memory: 0 byte, not used  
• Network: 100baseTx-FD (full duplex)  
• OS: Linux SuSE 10.0  
• OS Kernel: 2.6.13-15.12-smp, x86_64 GNU/Linux

It is noted, that all required files (audio files, program binaries, configurations, scripts, etc.) are read and write on the local hard disk. It means, that the slower network connection is not used to transmit any data. Furthermore, it is not allowed for other users to run other programs during the evaluation tests on the computer.

**Test Scenario to Achieve Test Goals \( \mathbb{I}_1 \) and \( \mathbb{I}_2 \)**

The theoretical scenario from page 55 with the embedding profile definition \( P_{E-\text{PerceptualHash}} \) is used for the description of the test scenario. The parameters of this profile are as follows. The parameter “in–signal” is set to the selected audio signal \( S \) out of the audio test set \( S \). “out–signal” are the marked audio signals after embedding the digital audio watermarks. The parameter “alg” is the selected watermark scheme (selected from \( \Omega \text{LSB}, \Omega 2\text{A2W}, \Omega \text{MS}, \Omega \text{SS}, \Omega \text{PM} \)) with its required parameter set defined in “alg-opt”. The parameter “hashalg” defines the used perceptual hashing algorithm with its required parameter set defined with “hashalg-opt”. Thereby, the perceptual hashing algorithm [HKO01] is selected. A Matlab implementation of the perceptual hashing algorithm \( H \) for audio content identification introduced by Haitsma and Kalker in [HK03] is used for the evaluation tests as transparency quality measure. The algorithm downsamples the signal to 5.5 kHz and divides the downsampled audio signal in time domain into overlapping windows of size 2048 samples. For each of the windows the frequency domain representation of the signal is split into \( m = 32 \) frequency bands. By applying the feature extraction function described in [HK03] on each of the frequency bands a fingerprint block with values in \( \{0, 1\} \) is computed. The \( m \) fingerprint blocks for each window are then joined to a so called sub-fingerprint which represents the audio signal in the window considered. The required parameters of it are as follows:

• Frequency bands (\( m \)): 32  
• Frame length (\( n \)): 2048 (samples)  
• Overlap fraction: windowing with a Hanning window with an overlap factor of 31/32  
• Downsampling frequency: 5.5 kHz

The output of the hashing algorithm can be parameterized to either return on the x-axis the 32 frequency bands (frequency band view - test goal \( \mathbb{I}_1 \)) or the frame number (temporal behavior; time based view test goal \( \mathbb{I}_2 \)) of the signal, while the y-axis always denotes the fingerprint blocks. In the comparison of two perceptual hashes in the frequency band view (test goal \( \mathbb{I}_1 \)) the average, minimum and maximum absolute embedding transparencies are measured as \( \text{phdg}_{\text{E ave}}^{\text{fb}}, \text{phdg}_{\text{E min}}^{\text{fb}}, \text{phdg}_{\text{E max}}^{\text{fb}} \). When using the time based view (test goal \( \mathbb{I}_2 \)) the average, minimum and maximum absolute embedding transparencies are measured as \( \text{phdg}_{\text{E ave}}^{\text{frame}}, \text{phdg}_{\text{E min}}^{\text{frame}}, \text{phdg}_{\text{E max}}^{\text{frame}} \).
Figure 3.12 shows the test scenario. After embedding the digital audio watermark, the detection/retrieval function is used to verify the successfully embedding measured as WM_{res}.

Figure 3.12: Test Scenario to Evaluate the Embedding Transparency of Exemplary Selected Audio Watermarking Schemes

The following Table 3.14 summarizes the chosen embedding parameter sets of the selected digital audio watermarking schemes. Note, that \( \Omega_{\text{LSB}} \) is used four times with four different embedding parameter sets to evaluate the impact of the embedding parameters scrambling and error correction codes (therefore four combination of them exists) on the perceptual hash. If the algorithm does accept a user defined embedding message, the string \( m=\text{"UniversityOfMagdeburg"} \), which contains 21 bytes (or 168 bits), is used.

<table>
<thead>
<tr>
<th>( E )</th>
<th>Selected Embedding Parameters ( p_E \in \mathcal{P}_E )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Omega_{\text{LSB}}^1 )</td>
<td>( p_E = { \text{LSB}_k = \emptyset, \text{LSB}_c = 0, \text{LSB}_r = 0, \text{LSB}_t = 3, \text{LSB}_j = 9 } )</td>
</tr>
<tr>
<td>( \Omega_{\text{LSB}}^2 )</td>
<td>( p_E = { \text{LSB}_k = \emptyset, \text{LSB}_c = 1, \text{LSB}_r = 0, \text{LSB}_t = 3, \text{LSB}_j = 9 } )</td>
</tr>
<tr>
<td>( \Omega_{\text{LSB}}^3 )</td>
<td>( p_E = { \text{LSB}_k = 1234, \text{LSB}_c = 0, \text{LSB}_r = 0, \text{LSB}_t = 3, \text{LSB}_j = 9 } )</td>
</tr>
<tr>
<td>( \Omega_{\text{LSB}}^4 )</td>
<td>( p_E = { \text{LSB}_k = 1234, \text{LSB}_c = 1, \text{LSB}_r = 0, \text{LSB}_t = 3, \text{LSB}_j = 9 } )</td>
</tr>
<tr>
<td>( \Omega_{\text{2A2W}}^* )</td>
<td>( p_E = { 2\text{A2WC} = \text{BIN}, 2\text{A2WE} = \text{ZT} } )</td>
</tr>
<tr>
<td>( \Omega_{\text{SS}}^* )</td>
<td>( p_E = { SS_k = 1234, SS_l = 500, SS_h = 10000, SS_\alpha = 2, )</td>
</tr>
<tr>
<td></td>
<td>( SS_f = 8192, SS_t = 3 } )</td>
</tr>
<tr>
<td>( \Omega_{\text{VAWW}}^* )</td>
<td>( p_E = { V\text{AWW}_k = 1234, V\text{AWW}_t = 40, V\text{AWW}_s = 0.1 } )</td>
</tr>
<tr>
<td>( \Omega_{\text{PM}}^* )</td>
<td>( p_E = { P\text{M}_{k\text{pub}} } )</td>
</tr>
<tr>
<td>( \Omega_{\text{MS}}^* )</td>
<td>( p_E = { \emptyset } )</td>
</tr>
</tbody>
</table>

Table 3.14: Summarized Selected Embedding Parameters \( p_E \) of the Selected Watermarking Schemes.
The selected parameter sets for watermark detection/retrieval are corresponding to the selected embedding parameter sets. Therefore, for the digital audio watermarking schemes \( \Omega^1_\text{LSB} \), \( \Omega^2_\text{LSB} \), \( \Omega^3_\text{LSB} \), \( \Omega^4_\text{LSB} \), \( \Omega^\ast_{\text{AWW}} \), \( \Omega^\ast_{\text{SS}} \) and \( \Omega^\ast_{\text{MS}} \) the retrieval parameters are \( p_R = p_E \). For \( \Omega^\ast_{\text{VAWW}} \) the watermark detection parameter is set to \( p_D = P_E \), whereby for \( \Omega^\ast_{\text{PM}} \) the retrieval parameter is set to \( p_R = \{ P_{M_{\text{priv}}} \} \).

### 3.3 Test Results of Profile Based Evaluation of Digital Audio Watermark Schemes with the Application Profile Perceptual Hashing

In this section, the evaluation results of the application oriented digital audio watermark evaluation with the application scenario of perceptual hashing \( P_{E-\text{PerceptualHash}} \) defined in the evaluation methodology in section 3.1.5.3 are presented and discussed. Thereby, the evaluation results regarding the two defined test goals \( \mathbb{I}_1 \) and \( \mathbb{I}_2 \) and their test scenarios described in section 3.2.3 are discussed with an intra- and inter-algorithm evaluation and analysis for each watermarking scheme. The visualizations used to show the effect and impact of the digital audio watermark embedding on the perceptual hashing function and therefore their embedding transparency measure have for the intra-algorithm evaluation and analysis a different scale for each algorithm and parameter set. This is used to show slightly differences within the frequency bands (test goal \( \mathbb{I}_1 \)) and frames (test goal \( \mathbb{I}_2 \)) of one digital audio watermarking scheme. For the required length of the perceptual hash value to normalize the perceptual hash difference grade values (\( \text{phdg}_{E_{\text{ave}}} \), \( \text{phdg}_{E_{\text{min}}} \) and \( \text{phdg}_{E_{\text{min}}} \)) a length of \( |H| = 5615 \) for the 32 frequency bands is measured. Note, for the presentation of the intra-algorithm evaluation and analysis results, the same structure is used, beginning with a summarization of successfully embedding of the message, followed by a discussion about the evaluation results regarding test goal \( \mathbb{I}_1 \) and closing with the presentation of the evaluation test results regarding test goal \( \mathbb{I}_2 \).

#### Intra–Algorithm Evaluation

**Algorithm \( \Omega^1_\text{LSB} \):** This watermarking scheme is able to embed the complete message \( m \) successfully into all audio signals \( S \in \mathbb{S} \). The evaluation results of the evaluation of the embedding transparency are shown in Figure 3.13. Figure 3.13(a) visualizes the test results regarding test goal \( \mathbb{I}_1 \), Figure 3.13(b) represents the test results regarding test goal \( \mathbb{I}_2 \).

For test goal \( \mathbb{I}_1 \), the absolute minimum changes for one frequency band occurred in the second frequency band, with a value of \( \text{phdg}_{E_{\text{min}}} \rightarrow_{E_{\text{ave}}} = 0.996 \). The absolute maximum changes for one frequency band occurred in the 26th frequency band with a value of \( \text{phdg}_{E_{\text{max}}} \rightarrow_{E_{\text{ave}}} = 0.05 \).

Evaluating the absolute average changed perceptual hash values over the complete audio test set \( S \) and all frequency bands, then the results show an average changing of \( \text{phdg}_{E_{\text{ave}}} \rightarrow_{E_{\text{ave}}} = 0.865 \).

For test goal \( \mathbb{I}_2 \), the absolute minimum changed perceptual hash over single frames is measured with 1 (one) for 5608 frames. The absolute maximum changing is measured for one frame with a value of \( \text{phdg}_{E_{\text{max}}} \rightarrow_{E_{\text{ave}}} = 0.281 \). The minimum and the average perceptual hash difference grade values are measured with \( \text{phdg}_{E_{\text{min}}} \rightarrow_{E_{\text{ave}}} = 1.000 \) and \( \text{phdg}_{E_{\text{ave}}} \rightarrow_{E_{\text{ave}}} = 0.974 \) over the used audio test set \( S \).
Algorithm $\Omega^1_{\text{LSB}}$: This watermarking scheme is able to embed the complete message $m$ successfully into all audio signals $S \in \mathbb{S}$. The evaluation results of the evaluation of the embedding transparency are shown in Figure 3.14. Figure 3.14(a) visualizes the test results regarding test goal $\mathbb{I}_1$, Figure 3.14(b) represents the test results regarding test goal $\mathbb{I}_2$.

For test goal $\mathbb{I}_1$, the evaluation test results are identical to the test results computed for $\Omega^1_{\text{LSB}}$, the absolute minimum changes for one frequency band occurred in the second frequency band, with a value of $\text{phdg}_{\mathbb{E}}^{f_b} = 0.996$. The absolute maximum changes for one frequency band also occurred in the 26th frequency band with $\text{phdg}_{\mathbb{E}}^{f_b} = 0.050$. Evaluating the absolute average changed perceptual hash values over the complete audio test set $\mathbb{S}$ and all frequency bands, then the results show an average changing of $\text{phdg}_{\mathbb{E}}^{f_b} = 0.865$.

For test goal $\mathbb{I}_2$, the absolute minimum changed perceptual hash over single frames is measured with 1 (one) for 5610 frames. The absolute maximum changing is measured for one frame times with a value of $\text{phdg}_{\mathbb{E}}^{f_b} = 0.312$. The minimum and the average perceptual hash difference grade values are measured with $\text{phdg}_{\mathbb{E}}^{f_b} = 1.000$ and $\text{phdg}_{\mathbb{E}}^{f_b} = 0.925$ over the used audio test set $\mathbb{S}$.
**Algorithm Ω_{LSB}^{\ast 3}:** This watermarking scheme is able to embed the complete message $m$ successfully into all audio signals $S \in S$. The evaluation results of the evaluation of the embedding transparency are shown in Figure 3.15. Figure 3.15(a) visualizes the test results regarding test goal $\Pi_1$, Figure 3.15(b) represents the test results regarding test goal $\Pi_2$.

For test goal $\Pi_1$, the absolute minimum changes for one frequency band occurred in the second frequency band, with a value of $\text{phdg}_{E_{\text{min}}}^{\text{fb}} = 0.998$. The absolute maximum changes for one frequency band occurred in the 30th frequency band with $\text{phdg}_{E_{\text{max}}}^{\text{fb}} = 0.317$. Evaluating the absolute average changed perceptual hash values over the complete audio test set $S$ and all frequency bands, then the results show an average changing of $\text{phdg}_{E_{\text{ave}}}^{\text{fb}} = 0.917$.

For test goal $\Pi_2$, the absolute minimum changed perceptual hash over single frames is measured to be 1 (one) for 5609 frames. The absolute maximum changing for single frames is measured four times with a value of $\text{phdg}_{E_{\text{max}}}^{\text{frame}} = 0.344$. The minimum and the average perceptual hash difference grade values are measured with $\text{phdg}_{E_{\text{min}}}^{\text{frame}} = 1.000$ and $\text{phdg}_{E_{\text{ave}}}^{\text{frame}} = 0.956$ over the used audio test set $S$.

![Figure 3.15: Evaluation Results of Ω_{LSB}^{\ast 3} within the Application Scenario of Perceptual Hashing](image)

**Algorithm Ω_{LSB}^{\ast 4}:** This watermarking scheme is able to embed the complete message $m$ successfully into all audio signals $S \in S$. The evaluation results of the evaluation of the embedding transparency are shown in Figure 3.16. Figure 3.16(a) visualizes the test results regarding test goal $\Pi_1$, Figure 3.16(b) represents the test results regarding test goal $\Pi_2$.

For test goal $\Pi_1$, the absolute minimum changes for one frequency band occurred in the first frequency band, with a value of $\text{phdg}_{E_{\text{min}}}^{\text{fb}} = 0.998$. The absolute maximum changes for one frequency band occurred in the 30th frequency band with $\text{phdg}_{E_{\text{max}}}^{\text{fb}} = 0.340$. Evaluating the absolute average changed perceptual hash values over the complete audio test set $S$ and all frequency bands, then the results show an average changing of $\text{phdg}_{E_{\text{ave}}}^{\text{fb}} = 0.917$.

For test goal $\Pi_2$, the absolute minimum changed perceptual hash over single frames is measured to be 1 (one) for 5609 frames. The absolute maximum changing for single frames is measured two times with a value of $\text{phdg}_{E_{\text{max}}}^{\text{frame}} = 0.311$. The maximum and the average perceptual hash difference grade values are measured with $\text{phdg}_{E_{\text{min}}}^{\text{frame}} = 1.000$ and $\text{phdg}_{E_{\text{ave}}}^{\text{frame}} = 0.957$ over the used audio test set $S$. 
**Algorithm $\Omega_{2A2W}^*$**: This watermarking scheme is able to embed the complete message $m$ successfully into all audio signals $S \in S$. The evaluation results of the evaluation of the embedding transparency are shown in Figure 3.17. Thereby, Figure 3.17(a) visualizes the test results regarding test goal $\mathbb{G}_1$, Figure 3.17(b) represents the test results regarding test goal $\mathbb{G}_2$.

For test goal $\mathbb{G}_1$, the absolute minimum changes for one frequency band occurred in the first frequency band, which a value of $\text{phdg}_{E_{\text{min}}}^{\text{fb}} = 0.981$. The absolute maximum changes for one frequency band occurred in the 9th frequency band with $\text{phdg}_{E_{\text{max}}}^{\text{fb}} = 0.354$. Evaluating the absolute average changed perceptual hash values over the complete audio test set $S$ and all frequency bands, then the results show an average changing of $\text{phdg}_{E_{\text{ave}}}^{\text{fb}} = 0.815$.

For test goal $\mathbb{G}_2$, the absolute minimum changed perceptual hash over single frames is measured with 1 (one) for 5129 frames. The absolute maximum changing is measured for one frame with a value of $\text{phdg}_{E_{\text{max}}}^{\text{frame}} = 0.031$. The minimum and the average perceptual hash difference grade values are measured with $\text{phdg}_{E_{\text{min}}}^{\text{frame}} = 0.996$ and $\text{phdg}_{E_{\text{ave}}}^{\text{frame}} = 0.848$ over the used audio test set $S$. 

Figure 3.17: Evaluation Results of $\Omega_{2A2W}^*$ within the Application Scenario of Perceptual Hashing
Algorithm $\Omega^{\ast}_{SS}$: This watermarking scheme is not able to embed the complete message $m$ successfully into all audio signals $S \in S$. Thereby, into the audio file $f r e r 07 . l$ it was not able to embed any character. For all other audio files $S \in S$, it is six times able to embed the substring “Unive”, three times “Univer”, two times “Univers” three times “Universi” and one times “Universit”. The evaluation results of the evaluation of the embedding transparency are shown in Figure 3.18. Thereby, Figure 3.18(a) visualizes the test results regarding test goal $\Pi_1$, Figure 3.18(b) represents the test results regarding test goal $\Pi_2$. For test goal $\Pi_1$, the absolute minimum changes for one frequency band occurred in the 31th frequency band, with a value of $\text{phdg}^\text{fb}_{E_{min}} = 0.875$. The absolute maximum changes for one frequency band occurred in the 24th frequency band with $\text{phdg}^\text{fb}_{E_{max}} = 0.166$. Evaluating the absolute average changed perceptual hash values over the complete audio test set $S$ and all frequency bands, then the results show an average changing of $\text{phdg}^\text{fb}_{E_{ave}} = 0.702$. For test goal $\Pi_2$, the absolute minimum changed perceptual hash over single frames is measured with 1 (one) for 2103 frames. The absolute maximum changing is measured for eleven frames with a value of $\text{phdg}^\text{frame}_{E_{max}} = 0.219$. The minimum and the average perceptual hash difference grade values are measured with $\text{phdg}^\text{frame}_{E_{min}} = 0.0970$ and $\text{phdg}^\text{frame}_{E_{ave}} = 0.778$ over the used audio test set $S$.

![Evaluation Results for $\Omega^{\ast}_{SS}$](image1)

(a) Frequency Band Focused Evaluation of $\Omega^{\ast}_{SS}$ for Test Goal $\Pi_1$

![Evaluation Results for $\Omega^{\ast}_{SS}$](image2)

(b) Frame Focused Evaluation of $\Omega^{\ast}_{SS}$ for Test Goal $\Pi_2$

Figure 3.18: Evaluation Results of $\Omega^{\ast}_{SS}$ within the Application Scenario of Perceptual Hashing

Algorithm $\Omega^{\ast}_{V A W W}$: This watermarking scheme does not require an embedding message. The embedding of the zero bit watermark was successful performed for all audio signals $S \in S$. The test results of the evaluation of the embedding transparency are shown in Figure 3.19. Thereby, Figure 3.19(a) visualizes the test results regarding test goal $\Pi_1$, Figure 3.19(b) represents the test results regarding test goal $\Pi_2$. For test goal $\Pi_1$, the absolute minimum changes for one frequency band occurred in the third frequency band, with a value of $\text{phdg}^\text{fb}_{E_{min}} = 0.545$. The absolute maximum changes for one frequency band occurred in the 13th frequency band with $\text{phdg}^\text{fb}_{E_{max}} = 0.005$. Evaluating the absolute average changed perceptual hash values over the complete audio test set $S$ and all frequency bands, then the results show an average changing of $\text{phdg}^\text{fb}_{E_{ave}} = 0.402$. For test goal $\Pi_2$, the absolute minimum changed perceptual hash over single frames is measured to be 1 (one) for 845 frames. The absolute maximum changing is measured for one frame with a value of $\text{phdg}^\text{frame}_{E_{max}} = 0.125$. The minimum and the average perceptual hash
difference grade values are measured with $\text{phdg}_{\text{Emin}} = 0.712$ and $\text{phdg}_{\text{Eave}} = 0.538$ over the used audio test set $S$.

![Image](image.png)

(a) Frequency Band Focused Evaluation of $\Omega_{VAWW}$ for Test Goal $\mathbb{I}_1$

(b) Frame Focused Evaluation of $\Omega_{VAWW}$ for Test Goal $\mathbb{I}_2$

Figure 3.19: Evaluation Results of $\Omega_{VAWW}$ within the Application Scenario of Perceptual Hashing

**Algorithm $\Omega_{PM}$:** This watermarking scheme is able to embed the complete message $m$ successfully into all audio signals $S \in S$. The evaluation results of the evaluation of the embedding transparency are shown in Figure 3.20. Thereby, Figure 3.20(a) visualizes the test results regarding test goal $\mathbb{I}_1$, Figure 3.20(b) represents the test results regarding test goal $\mathbb{I}_2$.

For test goal $\mathbb{I}_1$, the absolute minimum changes for singular frequency bands occurred in the second, fourth, 11th, 16th, 17th, 18th, 20th, 21th, 26th, 27th, 28th and 29th frequency band, with a value of $\text{phdg}_{\text{fbEmin}} = 1.000$. The absolute maximum changes for one frequency band occurred in the 19th frequency band with $\text{phdg}_{\text{fbEmax}} = 0.995$. Evaluating the absolute average changed perceptual hash values over the complete audio test set $S$ and all frequency bands, then the results show an average changing of $\text{phdg}_{\text{Eave}} = 0.998$.

For test goal $\mathbb{I}_2$, the absolute minimum changed perceptual hash over single frames is measured to be 1 (one) for 5607 frames. The absolute maximum changing for single frames is measured two times with a value of $\text{phdg}_{\text{frEmin}} = 0.344$. The minimum and the average perceptual hash difference grade values are measured with $\text{phdg}_{\text{frEmin}} = 1.000$ and $\text{phdg}_{\text{frEave}} = 1.000$ over the used audio test set $S$. 
Algorithm $\Omega_{MS}^*$: This watermarking scheme is not able to embed the complete message $m$ or parts of $m$ successfully into all audio signals $S \in \mathcal{S}$. Because of the evaluation goals ($\mathbb{I}_1$ and $\mathbb{I}_2$), where the embedding transparency is evaluated only, the worse retrieval capacity is for these evaluation tests neglected. The evaluation results of the evaluation of the embedding transparency are shown in Figure 3.21. Thereby, Figure 3.21(a) visualizes the test results regarding test goal $\mathbb{I}_1$, Figure 3.21(b) represents the test results regarding test goal $\mathbb{I}_2$.

For test goal $\mathbb{I}_1$, the absolute minimum changes is measured within the 22th frequency band with a value of $\text{phdg}_{fb}^{\text{min}} = 0.998$. The absolute maximum changes for one frequency band occurred in the 21th frequency band with $\text{phdg}_{fb}^{\text{max}} = 0.444$. Evaluating the absolute average changed perceptual hash values over the complete audio test set $\mathcal{S}$ and all frequency bands, then the results show an average changing of $\text{phdg}_{fb}^{\text{ave}} = 0.833$.

For test goal $\mathbb{I}_2$, the absolute minimum changed perceptual hash over single frames is measured to be 1 (one) for 5607 frames. The absolute maximum changing for single frames is measured two times with a value of $\text{phdg}_{frame}^{\text{max}} = 0.187$. The minimum and the average perceptual hash difference grade values are measured with $\text{phdg}_{frame}^{\text{min}} = 1.000$ and $\text{phdg}_{frame}^{\text{ave}} = 0.882$ over the used audio test set $\mathcal{S}$.

Figure 3.21: Evaluation Results of $\Omega_{MS}^*$ within the Application Scenario of Perceptual Hashing
Inter–Algorithm Evaluation

After presenting the intra–algorithm evaluation and analysis evaluation results of the selected watermarking schemes with their parameter sets, the inter–algorithm evaluation and analysis is presented and discussed here. Thereby, the evaluation test results are summarized and compare the digital audio watermarking schemes to each other as shown in Table 3.15, where the best and worst results are highlighted.

<table>
<thead>
<tr>
<th>$E$</th>
<th>Test Goal $\Pi_1$</th>
<th>Test Goal $\Pi_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\text{phdg}^\text{in}<em>{E</em>{\text{ave}}}$</td>
<td>$\text{phdg}^\text{in}<em>{E</em>{\text{min}}}$</td>
</tr>
<tr>
<td>$\Omega^1_{\text{LSB}}$</td>
<td>0.865</td>
<td>0.996</td>
</tr>
<tr>
<td>$\Omega^2_{\text{LSB}}$</td>
<td>0.865</td>
<td>0.996</td>
</tr>
<tr>
<td>$\Omega^3_{\text{LSB}}$</td>
<td>0.917</td>
<td>0.998</td>
</tr>
<tr>
<td>$\Omega^2_{\text{A2W}}$</td>
<td>0.815</td>
<td>0.981</td>
</tr>
<tr>
<td>$\Omega^2_{\text{SS}}$</td>
<td>0.702</td>
<td>0.875</td>
</tr>
<tr>
<td>$\Omega^2_{\text{VWW}}$</td>
<td>0.402</td>
<td>0.545</td>
</tr>
<tr>
<td>$\Omega^2_{\text{PM}}$</td>
<td>0.998</td>
<td>1.000</td>
</tr>
<tr>
<td>$\Omega^2_{\text{MS}}$</td>
<td>0.833</td>
<td>0.996</td>
</tr>
</tbody>
</table>

Table 3.15: Summarized Evaluation Results for Inter–Algorithm Evaluation and Analysis with a Fixed Capacity over the Complete Audio Test Set $S$ (with 5615 frames and 32 frequency bands)

By comparing the application depended evaluation results for the digital audio watermarking algorithms and focusing on $\Omega^*_{\text{LSB}}$ with its different used parameter sets the selection and the impact of $p_E$ are for test goal $\Pi_1$ as follows: If the digital audio watermarking scheme works in scramble mode by using a secret key $\text{LSB}_{k_1}$, then the impact of the perceptual hash is smaller than in the case of disabled scrambling mode (disabled scrambling mode $\text{phdg}^\text{fb}_{E_{\text{ave}}}=0.865$ and enabled scrambling mode $\text{phdg}^\text{fb}_{E_{\text{ave}}}=0.917$). Furthermore, the average changed values over the frequency bands ($\text{phdg}^\text{fb}_{E_{\text{ave}}}$) of the perceptual hash show a slight but monotonous increase if the scrambling mode is enabled while it shows an increase strongly alternating around the average value if the scrambling mode is disabled (compare Figures 3.13(a) and 3.14(a) with Figure 3.15(a) and 3.16(a)). In addition, the minimum number of changed perceptual hash values ($\text{phdg}^\text{fb}_{E_{\text{min}}}$) is higher if scrambling mode is disabled, see Table 3.15.

For test goal $\Pi_2$ the measured minimum values are equal for disabled and enabled scrambling mode ($\text{phdg}^\text{frame}_{E_{\text{min}}}=1.000$). The average and minimum phdg values differ, whereby $\text{phdg}^\text{frame}_{E_{\text{ave}}} \approx 0.926$ for an embedding parameter sets with disabled scrambling mode and $\text{phdg}^\text{frame}_{E_{\text{ave}}} \approx 0.956$, if the scrambling mode is enabled. It shows, that an enabled scrambling mode increases slightly the average embedding transparency if the frame based focus on the perceptual hash is used. For both, the minimum and average measured embedding transparency, there is no impact on an enabled or disabled error correction code identified. For the maximal measured phdg values, no significant impact of the scrambling mode nor error
correction code can be seen.

By focusing on the test goals $\mathbb{I}_1$ and $\mathbb{I}_2$ and comparing all watermarking schemes to each other, a digital audio watermark embedding with $\Omega^*_\text{PM}$ has the smallest impact (marked with light gray in Table 3.15) on the corresponding perceptual hash with an average changing of $\text{phdg}_{E_{\text{ave}}}^{fb} = 0.998$ and $\text{phdg}_{E_{\text{ave}}}^{\text{frame}} = 1.000$. In contrast, $\Omega^*_\text{VWW}$ provides the worst average embedding transparency with $\text{phdg}_{E_{\text{ave}}}^{fb} = 0.402$ and $\text{phdg}_{E_{\text{ave}}}^{\text{frame}} = 0.538$ over the complete audio test set $S$ and both point of views, frame and frequency band.

Within the frame based evaluation (test goal $\mathbb{I}_2$), a special characteristic for the digital audio watermarking scheme $\Omega^*_\text{PM}$ identified. This algorithm embeds non-redundantly in a linear way and show significant better results than the other algorithms, due to the fact that a large part of the audio signals is not marked. This fact can be very good observed for $\Omega^*_\text{PM}$ in Figure 3.20(b).

In the comparison of all algorithms for test goal $\mathbb{I}_2$, for the algorithms $\Omega^*_\text{LSB}$, $\Omega^*_\text{SS}$ and $\Omega^*_\text{VWW}$ a potential influence of the content of the audio signal can be seen. About half of the files in the test set start and end with a section of silence. This fact might result in the characteristic behavior of the $\text{phdg}_{E_{\text{frame}}}^{\text{frame}}$ at the beginning and the end of the signals for $\Omega^*_\text{LSB}$, $\Omega^*_\text{LSB}$ and $\Omega^*_\text{LSB}$ (see, for example, Figure 3.13(b)) and the behavior of the $\text{phdg}_{E_{\text{frame}}}^{\text{frame}}$ at the beginning and the end of the signals for $\Omega^*_\text{SS}$ and $\Omega^*_\text{VWW}$, see Figure 3.18(b) and Figure 3.19(b).

The following Figure 3.22 compares directly the embedding transparencies for test goal $\mathbb{I}_1$ in frequency band view (see subfigure 3.22(a)) as well as for test goal $\mathbb{I}_2$ in time based view (see subfigure 3.22(b)). The results highlight the fact, that $\text{phdg}_{E_{\text{ave}}}^{fb}$, $\text{phdg}_{E_{\text{ave}}}^{\text{frame}}$ and $\text{phdg}_{E_{\text{ave}}}^{\text{frame}}$ for $\Omega^*_\text{PM}$ are close to one, which implies the best performance for the selected application scenario. In contrast $\Omega^*_\text{VWW}$ has very low results for $\text{phdg}_{E_{\text{ave}}}^{fb}$ and $\text{phdg}_{E_{\text{ave}}}^{\text{frame}}$ in comparison with the other algorithms and this implies, that $\Omega^*_\text{VWW}$ has the worst transparency in the selected application scenario with the selected parameter sets.

When comparing the embedding domains of the selected audio watermarking algorithms in Figure 3.22 ($\Omega^*_\text{LSB}$, $\Omega^*_\text{PM}$ embedding in time domain, $\Omega^*_\text{2A2W}$ and $\Omega^*_\text{VWW}$ in wavelet domain, and $\Omega^*_\text{SS}$ and $\Omega^*_\text{MS}$ in frequency domain) no direct connection between the choice of the embedding domain and the computed embedding transparencies can be determined for the results given here.

![Figure 3.22: Inter–Algorithm Evaluation and Analysis for all Selected Digital Audio Watermarking Schemes](image-url)
3.4 Conclusion and Outlook

Derived from the variety of existing digital audio watermarking schemes and existing application scenarios with different security, audio quality and watermarking property requirements, a theoretical framework, useable for black box test and open for white box evaluation, has been designed and introduced. Thereby, the seven main properties transparency, capacity, robustness, complexity, verification, invertibility and security of digital audio watermarking schemes have been clearly defined and their measurements have been introduced. Based on these properties, evaluation profiles with different points of views of the users have been defined and created. Depending on the user of digital watermarks, it was possible to evaluate and analyze the single properties of a digital audio watermarking scheme with basic profiles or if a specific application scenario is given, then applications profiles support the decision and recommendation of given watermarking schemes. The defined and formalized watermark properties with their measurements are open for other types of media like image, video or 3D and they can be easily adapted. For example, the audio signal \( S \in \mathbb{S} \) could be replaced for a video signal with \( V \in \mathbb{V} \). In addition, a large audio test set, with a classification of the audio content has been designed, has been used for tests and can be used as motivation to create test sets for images, video or 3D with a content specific classification. With a well structured and classified test set, a content dependency of watermark properties can be identified.

Practical evaluation tests have been introduced the usage of the theoretical framework and the evaluation profiles. Selected audio watermarking schemes working in time, frequency and wavelet domain are evaluated with the application profile of perceptual hashing. It is shown, that the digital audio watermarking scheme \( \Omega^{\ast}_{2A2W} \) has the worst and \( \Omega^{\ast}_{PM} \) the best evaluation results with the selected parameter and test set. Note, if the requirements, for example, on robustness, are also in focus, then the evaluation results can be different and other watermarking algorithm recommendations can be given.

The introduced theoretical framework and the introduced profile based evaluation methodology of digital audio watermarking schemes in are meant to motivate and stimulate other researcher and developer to use and work with them. The adaptation of these measurements and the methodology as well as the profile based evaluation approach definitions could be enhanced to create and define new evaluation measurements especially for different application scenarios. Thus, it is possible to provide objectively comparability of the evaluated digital audio watermark schemes. Furthermore, the defined measurements in the theoretical framework could be used for security related application scenarios and the theoretical framework can be adapted for different types of media like image, video or 3D.
Chapter 4

Steganalytical Results for a Steganographic Algorithm

Evaluating the security of steganographic algorithms requires applying known steganalytical methods. According to commonly applied attack philosophies, various classifiers should be considered in this step. The necessary number of test images strongly depends on the number of features analysed by the applied classifiers. Within this report, we mainly focused on evaluating the security of the steganographic algorithm ECAP including comparisons to other steganographic schemes. Since it is not trivial to establish a test set of sufficient size due to the characteristics of ECAP, we applied two approaches to perform the analysis on the available set of test images: reducing the size of the images and reducing the size of the feature vector of the respective classifier. The report summarises the results of these tests pointing out that both approaches deliver reasonable results and can be used for comparisons. Our results have shown that ECAP provides better security than the other steganographic algorithms considering additional requirements on the cover images.

4.1 Introduction

Steganography is a method for confidential communication that protects not only the content of a secret message but hides even its mere existence. A steganographic algorithm embeds the secret message $emb$ into inconspicuously looking $cover$ data. The resulting $stego$ data should not be distinguishable from steganographically unused data, i.e., steganographic algorithms aim at producing plausible stego data.

The goal of steganalysis is to detect whether an intercepted object contains embedded messages, i.e., whether it was produced by a steganographic system. Thereby, steganalytical methods exploit characteristic traces caused by embedding. Steganalytical approaches can be divided into two main groups: Targeted algorithms are designed for a specific steganographic algorithm or embedding strategy while blind algorithms are independent from the embedding method. Targeted algorithms may achieve a higher accuracy in detecting the corresponding steganographic methods. Blind algorithms, on the other hand, allow for successfully evaluating a broad range of steganographic algorithms including even new ones.
Steganalysis usually starts with calculating a number of features – summarised in a feature vector – from the data to be analysed. Based on the values of these features, the data under investigation is classified as stego or as cover. The more features are evaluated, the more test data are required.

According to the goal of steganography, the security of steganographic schemes refers to the undetectability of the embedded messages. A steganographic algorithm is considered to be secure in practice, if there is no steganalytical method that allows for correctly classifying stego data produced by this algorithm with a probability better than random guessing.

Evaluating the security of a new steganographic technique should adhere to commonly accepted guidelines as summarised in [CMB+08]. The analysis should consider currently known blind methods and work on a large and diverse set of test data. As already mentioned, the size of the feature vector influences the size of the test set. However, the requirement for a large test set might be difficult to realise in a concrete scenario.

This report summarises the steganalytical evaluation of the steganographic algorithm ECAP (Embedding Considering Adjacent Pixels) introduced in [Fra08]. First steganalytical results already have been presented in [Fra08], this report bases mainly on the additional analysis in [Rö08]. The algorithm ECAP uses scanned grey scale images as cover data. It first evaluates a number of scans to derive a model which is then used for embedding by generating a stego image that contains the secret message. This processing implies that providing a large test set is a time consuming task. Therefore, we have tested various strategies to reduce the effort within our tests. The main approaches are reducing the image size and reducing the size of the feature vector of the respective classifier. The evaluation also considers comparisons to other steganographic schemes, namely ±1 steganography, e.g., [Sha01], Stochastic Modulation [FG03] and a possible realisation of Perturbed Quantization [FGS04a]. Including other algorithms additionally enables evaluating the approaches to reduce the effort since the other algorithms do not require a number of scans for each stego image.

Section 4.2 shortly introduces the steganographic algorithms considered in our tests. We move on to an overview on the selected classifiers in Section 4.3. Section 4.4 presents the practical results of our tests and Section 4.5 summarises and gives an outlook.

### 4.2 Steganographic Algorithms

#### 4.2.1 Embedding Considering Adjacent Pixels (ECAP)

The general goal of the algorithm is to generate a plausible image based on a model derived by analysing a number of realisations of the cover image. Particularly, ECAP aims at considering dependencies between pixels, motivated by the known fact that pixels of a natural image are not stochastically independent [GW02]. The generated image contains the message to be embedded; the message bits are encoded into the least significant bits of the pixels.

As different realisations of one cover, ECAP uses a number of scans of that image. The scans are not identical due to the noise that is inherently present in any digitalisation process. Additional differences are caused by mechanical irregularities of the scanner: Even if the position of the analogue image on the scanners’ platen is not changed, the exact scan positions
of repeated scans will not be identical. Consequently, there will be differences between the pixels of the compared images especially on grey edges.

The algorithm utilises the different digital representations of the analogue image for generating a plausible stego image independent from the reason for the differences. It describes plausible values for each pixel to be generated by means of conditional probabilities calculated from a set of adjacent pixels. The set of adjacent pixels needs to be defined as a parameter; best results could be achieved by evaluating the direct neighborhood of a pixel. Generating a new pixel requires that the considered adjacent pixels are already fixed. Consequently, the first line and the first row have to be initialised before embedding. Initialisation simply copies the pixel values from the first scan.

For each of the adjacent pixels, ECAP generates a suggestion for a plausible pixel value; the final suggestion is than calculated as weighted mean of the single suggestions. The weights for the single suggestions control the influence of the adjacent pixels. Embedding is done by rounding the final suggestion to the nearest integer value that represents the next message bit to be embedded.

During the evaluation of the initial version of ECAP, improvements have been suggested [R08]. Generally, these improvements deal with problems caused by unwanted differences between the used scans, introduced, e.g., by irregular shifts. The new version mainly considers two aspects: It improves the initialisation by first evaluating which of the scans is suited best and it applies a dynamic selection of scans for determining the conditional probabilities for each pixel including also dynamic weights.

The embedding rate of ECAP is influenced by the characteristics of the scans as well as by the parameters used for the dynamic selection. The result of the dynamic selection depends on the pixel values at the evaluated position and, hence, on the number of scans used for the evaluation. In case a message does not require to use the whole embedding capacity, the remaining pixels are generated according to the conditional probabilities.

4.2.2 ±1 Steganography or LSB Matching (LSBM)

We also evaluated further steganographic algorithms to enable assessing the security of ECAP. The selected algorithms should work similar to ECAP to allow for a reasonable comparison. Thus, we considered algorithms embedding into uncompressed images. The selected algorithms only need one cover for generating one stego image, hence, they could be used for evaluating the different approaches for steganalysis in case of a small test set.

±1 Steganography or LSB Matching (LSBM) [Sha01] directly represents the message bits in the least significant bits of the cover image. If the least significant bit of the pixel already represents the next message bit to be embedded, it remains unchanged. Otherwise, the algorithm adds or subtracts 1 with equal probability. Thus, only small changes are introduced.

Since LSBM does not evaluate the cover pixel before embedding, it can achieve an embedding rate of 100%. We simulated LSBM for our tests according to the required embedding rate spreading the modifications randomly over the cover if necessary.
4.2.3 Stochastic Modulation (StM)

Stochastic Modulation (StM) [FG03] embeds a secret message by adding a random stego noise signal to the pixels of the cover image. The distribution of the stego noise signal can be chosen arbitrarily, e.g., the signal could adhere to a Gaussian distribution rounded prior to embedding. If the stego noise signals equals zero, the next bit is not used for embedding. Embedding and extracting message bits bases on a parity function applied to the pixels and the stego noise signal. The result of this function represents the message bit; if the result equals the message bit, the value of the stego noise signal is added to the pixel, otherwise, it is subtracted. An enhanced version of Stochastic Modulation works with two stego noise signals.

We have used the simple variant within our tests considering a zero mean Gaussian signal. The embedding rate of this variant is determined by the parameters of the stego noise signal, more precisely, by the number of zeros of the stego noise signal. The algorithm can achieve an embedding rate of approximately 0.8 bit per pixel (bpp) [FG03]. The smaller the variance, the smaller the embedding rate; a stego noise signal with a variance equal to 1 implies an embedding rate of about 0.61 bpp. The necessary modifications are spread over the cover image according to single values of the stego noise signal.

4.2.4 Perturbed Quantization (PQ)

Perturbed Quantization (PQ) [FGS04a] utilises information about the cover data known only to the sender. The approach is based on Wet Paper Codes that allow for embedding without the necessity that sender and recipient share the selection channel. Consequently, the sender can freely choose exactly the pixels he wants to use for embedding and, hence, he can restrict modifications to the best suited pixels.

PQ provides a possibility to determine such suited pixels during an information-reducing process like scaling down an image or reducing its color depth. In both cases, the values for the new pixels are real numbers and still need to be “quantised”. This quantisation step is slightly changed so that the least significant bits of the pixels represent the message bits. To ensure that embedding does not significantly influences the quantisation, only values within a narrow interval around 0.5 are changed at all. These values establish the changable values utilised by Wet Paper Codes.

As information reducing process, we used scaling down images by a factor of 2. We simulated the embedding process, i.e., we modified changable pixels according to a random stego signal without implementing Wet Paper Codes. To ensure good results of this embedding technique, real pixel values were sorted according to their deviation from 0.5 and used for embedding in this order. The embedding rate of PQ depends on the characteristics of the pixel values after scaling down; again, we worked with comparable rates within our tests.
4.3 Steganalytical Methods used in the Tests

4.3.1 Rationale for Selection

Evaluating the security of ECAP required selecting suitable classifiers. We first focused on blind classifiers following the general guidelines for evaluating new steganographic algorithms. According to [GFH06], steganalysis can achieve best results if the features are calculated in the embedding domain. Consequently, we aimed at selecting steganalytical approaches working in the spatial domain. However, there are also other interesting classifiers considering, e.g., correlations between pixels. They are of special interest since ECAP especially aims at generating plausible correlations. Thus, we also considered classifiers calculating features, e.g., in the wavelet domain.

Additionally, we considered classifiers specialised on LSBM steganography, not only because LSBM is also used for comparison but also since we assume them to detect smallest steganographic manipulations. Due to the aforementioned reason, we selected classifiers working in the spatial domain.

Both blind and targeted classifiers must be able to work with never compressed grey scale images. With our choice of steganalytical methods we want to cover a broad range of features. Altogether, we used six classifiers; four of them are blind ones and two are tailored to LSBM steganography. In the following, we give a short overview on the selected classifiers, summarise the evaluated features and comment some small changes necessary for our analysis.

4.3.2 Selected Classifiers

Chen et al. introduced a blind classifier based on statistical analysis of empirical matrices (EM) [CWTG06]. An empirical matrix or co-occurrence matrix evaluates the frequency of color values or grey scales at different positions within an image described by a relation that defines distance and direction. Additionally, prediction error images are calculated by estimating the value of a new pixel from the surrounding pixels and calculating the error as difference to the actual grey scale at that point. Utilising these empirical matrices and the prediction error image, the authors calculate a 1-D (one dimensional) projection histogram for each matrix. Afterwards they take multiple order moments of the histograms themselves and of their discrete Fourier transformations (DFT), respectively, as features. In contrast to the authors which use support vector machines (SVM) for analysing the resulting feature vector (FV) we use Fisher Linear Discriminant (FLD). This modification enables us to achieve a better comparability with the other classifiers for which we use FLD as well.

A blind classifier which operates in the wavelet domain is proposed by Shi et al. [SGXG+05]. The authors calculate statistical moments of different orders from the DFTs of different wavelet subbands (Haar). They calculate these features both for the image and its prediction error image what results in a 78-D FV, thus, we call this classifier in the following Steg78. Again, we do not use the proposed method of the paper (neural network) for analysing the FVs but FLD to allow for a better comparability and to avoid the very time-consuming task to train the neural network.

Li et al. consider steganalysis as a texture classification problem [LHS08] and propose
a method called textural feature based universal steganalysis (TFBUS). They convolve an image with specific 2-D DCT masks for feature extraction, calculate the probability mass functions (PDF) of the result and take a certain amount of these PDFs as features. The authors analyse the resulting FV by means of FLD.

A further wavelet based blind classifier is proposed by Goljan et al. [GFH06], named Wavelet Absolute Moment steganalysis (WAM). The authors denoise the subbands of an image wavelet decomposition (8-tap Daubechies, quadrature mirror filter) by means of a maximum a posteriori (MAP) estimation and a Wiener filter, subtract the noise reduced subbands from the original subbands and get the noise residual of each considered subband. They take absolute central moments of these noise residuals as features and analyse the resulting FV with FLD.

Ker proposed a classifier for detecting LSBM steganography [Ker05]. He introduced two discriminators applying the histogram characteristic function center of mass (HCFCOM, proposed by Harmsen and Pearlman [HP03]) in a special way. The suggested discriminators are called Adjacency HCFCOM (A.HCFCOM) and Calibrated Adjacency HCFCOM (C.A.HCFCOM).

Liu et al. introduced another approach to detect LSB matching steganography: feature mining and pattern classification (FMPC) [LSCX08]. They use a broad set of features for their classifier. We only take the subset of correlation-features to restrict the required effort and analyse the resulting 83-D FV by means of FLD. The authors also tested several feature subsets showing that the correlation features are essential for the steganalytical performance.

### 4.3.3 Problem: Size of the Test Set

As mentioned above, a reasonable analysis requires a sufficient large test set. The actual required size of the test set strongly depends on the size of the FV calculated by the respective classifier and on the complexity of the classifier. Within our tests, we were geared to the test set sizes used in the papers the classifiers are presented in. Both HCFCOM and FMPC work with very huge databases; the remaining classifiers use test sets calculated from a number of cover images which is about six (EM) to sixteen (TFBUS) times as big as the evaluated FVs. These test set sizes are used especially in the respective training steps. Hence, we assume a factor of ten to be reasonable to get feasible results. Table 4.1 summarises the resulting sizes of the test sets for the introduced steganalytical algorithms according to the size of the respective FV.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Size of FV</th>
<th>Size of test set (number of images)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCFCOM [Ker05]</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>C.A.HCFCOM</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>EM [CWTG06]</td>
<td>108</td>
<td>1080</td>
</tr>
<tr>
<td>FMPC [LSCX08]</td>
<td>83</td>
<td>830</td>
</tr>
<tr>
<td>Steg78 [SGXG05]</td>
<td>78</td>
<td>780</td>
</tr>
<tr>
<td>TFBUS [LHS08]</td>
<td>110</td>
<td>1100</td>
</tr>
<tr>
<td>WAM [GFH06]</td>
<td>36</td>
<td>360</td>
</tr>
</tbody>
</table>

Table 4.1: Numbers of test images required by the selected classifiers.
There are up to 1100 images needed (TFBUS) to get feasible test results what implies problems for the realisation of the analysis considering ECAP: Due to the fact that ECAP requires \( N \) scans for generating one stego image, we cannot use existing image databases such as the databases used in the papers the classifiers are presented in. Generating a database of sufficient size would require a huge effort; if we want to use ten scans of an image as input for ECAP, we had to perform 11000 scans.

Thus, we worked with an already existing database. This available image database consists of 204 multiple scanned photographs (scanned using an HP ScanJet 6300C, applying a common resolution of 200 dpi). We splitted these images in two parts each to get 408 images of reasonable size (512 \( \times \) 512 pixel). Hence, all steganalytical algorithms with FVs larger then 40 are not applicable to our ECAP test set without problems.

We investigated two possible solutions for this problem. The first one is to split every available image into smaller ones to increase the total image number. The other one is to reduce the FVs of the problematic classifiers to a manageable size. We applied these strategies to images generated by the other steganographic algorithms to be able to compare them: Since the other steganographic algorithms require only one cover image for embedding, there is a possibility to extend the test set by single scans. After this comparison, we moved on to evaluate the security of ECAP comparing it to the other steganographic algorithms.

### 4.4 Practical Results

#### 4.4.1 Test Sets and Test Procedure

Generally, one needs two sets of test images for performing a classification, one for training the classifier and the other one for testing the defined classification settings for data not used during the training phase. Both sets contain cover images from the underlying database as well as stego images generated with the corresponding steganographic algorithm from these cover images. Within the tests summarised in this report, we only worked with the training set. Thus, the results presented in this report reflect the separability of the given test set into cover and stego images in the best case from point of view of steganalysis: One can expect that features calculated from previously not considered images might differ from the measured features, hence, detection in this case might be worse than for the training set.

During our investigations, we always compared different sets of stego images. These sets are generated either by different steganographic algorithms using the same embedding rates to ensure comparability or by one and the same algorithm using different embedding rates. Embedded messages are always random bit strings and different for every image. As already mentioned, we use FLD for all classifiers due to simplicity and a better comparability.

Diagrams showing the results of our tests refer to the accuracy (also called detection reliability) describing the performance of a classifier on a certain test. According to [Fri05], we used the normalised value

\[
\text{accuracy} = 2AUC - 1
\]
where \textit{AUC} is the area under the Receiver Operating Characteristic (ROC) curve. This value is also known as Gini coefficient \cite{Faw04}. An accuracy equal to 1 represents perfect separability while an accuracy equal to 0 corresponds to random guessing.

4.4.2 Preliminary Considerations

Issues Regarding Reduced Image Size

As already mentioned, we could use only a database of 408 images of size $512 \times 512$ pixels. Increasing the number of images becomes possible by splitting up every $512 \times 512$ image into four $256 \times 256$ non-overlapping portions. However, this processing should be evaluated to assess its influence on the results of the analysis: First, the features are calculated from less data since the images are smaller and, second, portions taken from one and the same image might be of similar characteristics.

Particularly, we wanted to know in which direction the classifier performance evolves due to the smaller size of the images.

Issues Regarding Reduced FVs

The existing database of 408 images can be used only for analysis applying the classifiers HCFCOM and WAM. Therefore, we investigated reducing the size of the feature vector as another possible solution. We generally expect a decreased classifier performance if less features are evaluated.

However, especially blind classifiers might not necessarily be affected significantly by such a reduction. They cover a broad range of features and there is the possibility that the remaining features are sufficient to reliably detect a specific steganographic algorithm. On the other hand, if improvements of a steganographic algorithm are tested with such reduced classifiers it might be possible that features relevant for the improved version are missing what should be considered in further steganalytical evaluations.

Issues Regarding Comparison with PQ

There is a further issue that needs to be considered in our tests. We simulated PQ as part of a resizing operation within our tests (Section 4.2.4). The resizing operation implies that we get smaller stego images without increasing their number - if we use the 408 images of size $512 \times 512$ pixel, we get 408 images of size $256 \times 256$ pixels. We do not want to split up the smaller images again since this results in really small images providing no reasonable results for realistic scenarios. Hence, for such comparisons more compact FVs are needed anyway.

Regarding a comparison with PQ, there is additionally the basic question of the impact of resizing the scanned, never compressed images prior to embedding. Thus, we also performed tests to assess the influence of this operation on the steganalytical results.
Basic Questions

To conclude, we can derive the following questions from the issues discussed above:

1. What is the impact of reducing the size of the test images considering using smaller portions out of one larger image and resizing?
2. What is the influence of reducing the FV applied by a classifier?
3. Is it more advantageous to work with a larger test set of smaller images or to work with a smaller test set of larger images and reduced FVs?
4. To which degree influences resizing prior to embedding the security of a steganographic algorithm?

To answer the first two questions we exemplarily investigated the influences of different settings on LSBM. This algorithm requires only one cover image as input, what allows for using additional images scanned only once. We extended the existing 408 image test set to a set of 1120 images of the same characteristics ($512 \times 512$ cutouts of each of the different photographs scanned with a resolution of 200 dpi). This extended test base enabled us to use all selected classifiers and to assess the possibilities for reducing the image size as well as for reducing the FVs.

4.4.3 Evaluating the Approaches for Reducing the Effort

4.4.3.1 Reduced Image Size

Based on the extended set of 1120 cover images of size $512 \times 512$ pixels, we generated five sets of 1120 cover images used for embedding as follows:

- use all existing cover images of size $512 \times 512$ for comparison,
- resize all 1120 images by a factor of two getting images of size $256 \times 256$,
- choose randomly one of the four non-overlapping $256 \times 256$ portions out of each image,
- take all eight non-overlapping $256 \times 256$ portions out of the two $512 \times 512$ cutouts of one photograph; use 140 randomly chosen photographs, and
- chose $256 \times 256$ pixel cutout from the center of each image.

The central part was also used as cutout in [LF05] to get rid of the low complexity parts of the image. Actually, we are mainly interested in possibilities for increasing the image number that allow for using multiple non-overlapping cutouts per image. However, we included this possibility to assess whether the other approaches imply a significant decrease of accuracy.

The five different sets of 1120 cover images were used to generate corresponding sets of stego images applying LSBM. We created two sets of stego images for each cover set: one by embedding with maximum embedding rate of 1 bpp and another one by embedding with a
more common embedding rate of 0.3 bpp. Altogether, this processing resulted in ten sets of stego images. We classified the corresponding cover and stego sets using all six classifiers to get a reference to compare with. Figure 4.1 and Figure 4.2 present the results of this test.

![Figure 4.1](image1.png)

**Figure 4.1:** Compare possibilities for reducing the image size (1 bpp).

![Figure 4.2](image2.png)

**Figure 4.2:** Compare possibilities for reducing the image size (0.3 bpp).

Generally, taking small cutouts of larger images tends to worse steganalytical results in comparison to analysing the larger images. We assume that this is mainly because of the less varying image content of such cutouts. As an example, consider the case that a portion of an image showing a natural scene contains only a large piece of heaven, i.e., a large low complexity region. Since the steganalytical results depend on the image size, evaluation of a steganographic algorithm should consider image sizes reasonable for the later use of the algorithm.

The influence of the resizing operation does not lower the classifier performance in any case. The results are significantly better compared to the results for cutouts of the same size. Resizing affects the noise inherently present in the images due to digitalisation what might
influence the steganalytical results. In case of embedding with maximum embedding rate, the accuracy gets even higher for classification with EM, Steg78, TFBUS, and WAM compared to the results of the large images. A more reasonable embedding rate of 0.3 bpp implies a decrease of accuracy except for EM where accuracy is higher in a marginal manner.

There are no significant differences in the classification results comparing the utilised methods for producing cutouts. Thus, the number of images can be increased by using multiple cutouts of one image keeping in mind the shrinking classifier performance when splitting up few large images to many small ones.

4.4.3.2 Reduced Feature Vectors

The investigations regarding reduced FVs are also based on the five sets of cover images and the resulting ten sets of stego images introduced in the previous section. Combining both approaches is necessary since we aim at comparing steganographic algorithms to PQ. We selected EM and TFBUS as examples for classifiers utilising large FVs based on previous investigations [Rö8].

There are several possibilities for feature selection, e.g., using ANOVA [AMS03]. A more detailed discussion can be found in [LSCX08]. However, we did not apply specific techniques for reducing the FVs. We rather utilised obvious possibilities derived from the structure of the FVs to exclude features or subgroups of features. The possibility to exclude the selected features was tested by means of empirical tests [Rö8].

Finally, we used only the defined empirical matrices with step size one (instead of one, two and three) for EM what implied a FV containing 36 features. The TFBUS classifier allows to choose a smaller value for the parameter $R$ specifying the observed window of the measured PMFs. Selecting $R = 1$ (instead of $R = 5$) leads to a FV of size 30. Altogether, we analysed each of the ten combinations of cover and stego sets four times, considering for each of the two classifiers the original and the reduced FV. Figure 4.3 and Figure 4.4 present the results of these tests.

![Figure 4.3: Compare original and reduced FVs (1 bpp).](image)
Reducing the FV decreases the accuracy of the classification. There is a decrease of about 0.08 for EM and 0.14 for TFBUS.

4.4.3.3 Conclusions

Considering the results of the investigations reported in this section, there is no possibility for definitely answering question 3, i.e., for deciding which of the approaches for reducing the effort should be preferred if providing a test set of sufficient size is difficult in practice. Both approaches usually imply a decreased accuracy; the concrete decrease depends on the classifier and the embedding rate and, additionally, on the test images. Actually, it is not possible to reduce the FV of any classifier without producing dramatically worse classification results; in such cases only analysis with reduced image size are possible.

Hence, we applied the approach of reducing the image size to be able to apply all of the introduced classifiers without reducing the size of the FV if possible. Thereby, the size of the images is reduced as less as possible to get more reasonable results.

However, solely reducing the image size is not possible for comparisons to PQ. There is a need to work with reduced FVs since we do not want to reduce the image size further, i.e., after the resizing step. Thus, we get results that can be used as comparison between the different algorithms but have to keep in mind that classification for larger images without reducing the FVs would work with a better accuracy.

4.4.4 Compare ECAP, LSBM, and StM

The investigations reported in this Section base on the test images of photographs scanned several times due to the requirements of ECAP. There are 204 different images of size 980×700 pixels that can be used to generate cutouts according to the conclusions from the previous investigations.

Following our defined guideline to use as big images as possible for steganalysis, we split...
up the photographs in $340 \times 340$ pixel portions. This splitting is not possible without any overlapping, but in fact there are only small overlapping areas (Figure 4.5). Altogether, we used six cutouts per image and, hence, the resulting test set contained 1224 cover images. This size of the test set is sufficient for all classifiers according to our defined guideline.

![Figure 4.5: Cutouts of $340 \times 340$ pixels generated from the photographs.](image)

Within these investigations, we used ECAP as reference algorithm with respect to the embedding rates. Due to the dependencies of the embedding rate (Section 4.2.1), there is a specific embedding rate for each image. We used two different parametrisations, one focusing on minimising the detectability of embedding, the other one focusing on maximising the embedding rate. The former approach lead to an average embedding rate of 0.34 bpp, the latter one to an average embedding rate of 0.57 bpp. We want to point out again that these embedding rates are average values only since the final embedding rates not only depend on the parameters but also on the characteristics of the cover image. Figure 4.6 shows the actual distribution of the embedding rates for the two parametrisations.

![Figure 4.6: Distribution of the actual embedding rates for the test set.](image)
We used exactly the same embedding rates per image as realised by ECAP for LSBM and StM. The results of the analysis are presented in Figure 4.7.

ECAP achieves the lowest accuracy in all test cases considered in these investigations. It might even be substantially better than LSBM and especially StM depending on the classifiers and the embedding rate. FMPC achieves best results in classifying ECAP for an average embedding rate of 0.34 bpp, EM for an average embedding rate of 0.58 bpp. Regarding the other steganographic algorithms, TFBUS yields the best accuracy for both LSBM and STM, its accuracy in classifying ECAP is approximately in the region of EM and FMPC.

### 4.4.5 Compare ECAP, LSBM, and PQ

Following our guideline to use as large images as possible, we did not use the $512 \times 512$ pixel cutouts as source material for the resize operation but two $680 \times 680$ pixel cutouts of the source photographs. The cutouts are positioned in the upper left and the upper right corner of the photographs, respectively. Hence, embedding with PQ produces $340 \times 340$ pixel images.

For comparisons to ECAP and LSBM, we generated two different sets of cover images: First, we used a randomly chosen $340 \times 340$ pixel cutout of each $680 \times 680$ source image and, second, we generated $340 \times 340$ pixel images by resizing the $680 \times 680$ cutouts of the photographs. Average embedding rates for ECAP result again from reasonable parametrisations; the actual embedding rates for the 408 images of the test set are summarised for two of the average embedding rates in Figure 4.8.

As in the investigations described in the Section 4.4.4, the embedding rates yielded by ECAP were used for generating the other stego images by applying LSBM and PQ, respectively. We could use only HCFCOMs, EM (reduced FV), TFBUS (reduced FV) and WAM within these tests due to the limited image number. Figure 4.9 and Figure 4.10 show the results of these investigations.

The classifiers can detect LSBM with a higher accuracy than PQ and ECAP. Generally, TFBUS achieved best accuracy in classification. Surprisingly, results of classifying stego
Figure 4.8: Distribution of the actual embedding rates for the test set.

Figure 4.9: Comparing ECAP, PQ and LSBM using cutouts.

Figure 4.10: Comparing ECAP, PQ and LSBM using resized images.
images generated by ECAP using resized images are approximately in the range of stego images generate by PQ or even better from point of view of steganography. As already mentioned, possible problems of ECAP may result from irregular shifts [R08]. We assume the resizing to mask such irregular shifts and, hence, complicate steganalysis.

4.5 Conclusion and Outlook

Within this report, we compared the security of different steganographic algorithms regarding a number of steganalytical methods. We mainly focused on analysing the steganographic algorithm ECAP what implied problems regarding the size of the test set. State of the art classifiers evaluate a large number of features implying the need for a large test set. However, ECAP requires a number of scans as different realisations of an image what further increases the number of cover images needed.

We investigated two possible solutions for this problem: reducing the image size and reducing the size of the feature vector of the respective classifier. As expected, the former approach results in decreased classification accuracy. Nevertheless, the results show that it is possible to split up few large images in smaller portions getting a sufficient number of images without falsifying the steganalytical results: Proportions between different steganographic algorithms or between stego images created by embedding using different parameters are maintained. Since the size of the images influences the steganalytical results, the test images should be as large as possible if reducing the size is necessary. Cutouts can be arbitrarily generated. In contrast, resizing before embedding influences the results of steganalysis and, furthermore, its influence depends on the embedding rate.

Finally, further tests confirmed the possibility to reduce the size of the FV. However, improved version of a steganographic algorithm should be carefully evaluated. There might be the case that formerly excluded features are now be necessary for good steganalytical results. Thus, reducing the FV should be evaluated again.

Based on these results, we compared ECAP, LSBM, StM applying an extended test set through splitting up the larger images and afterwards we compared ECAP, LSBM and PQ applying reduced FVs. ECAP achieved lowest accuracy values compared to the steganographic algorithms LSBM and StM. The comparison to PQ yielded the result that resizing the cover image significantly improves the security of ECAP. We assume that resizing reduces known problems with irregular shifts in the scanned images since it implies a lower resolution. ECAP achieved results similar to PQ if resized images are used as covers.

Finally, we want to point out the presented results are based on one test set; we assume that other test sets would deliver similar results. Future work needs to be done to continue evaluation based on enhanced versions of the algorithm ECAP. Another topic is to improve the feature selection by applying common strategies.
Chapter 5

Summary

This report summarizes the research activities of the WAVILA WP3, which is a virtual lab of the ECRYPT network of excellence in cryptology. The introduced theoretical and practical framework for digital audio watermarking evaluation and the steganalytical evaluation of marked objects are presented and discussed in this report. After a short summary of the work done in WAVILA WP3, the watermarking parameters are clearly defined for watermarking evaluation. For the watermark evaluation itself are so called profiles defined and formalized, which helps on one hand watermark designer with deep inside knowledge and on the other hand end-users with few inside knowledge of the watermark algorithm. A practical evaluation showed the usability of the theoretical framework on the example of application oriented evaluation within the field of perceptual hashing. It is shown, that application scenario depending new measurements can be defined to provide an objective comparison. Furthermore, exemplary selected steganographic algorithms for images are introduced and its detectability with steganalytical methods evaluated. The main focus is set on the embedding considering adjacent pixels (ECAP) algorithm evaluated with, for example, different image sizes.
Acknowledgements

The work about the measurement and profile definition described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.
ECRYPT — European NoE in Cryptology
Bibliography


[LD06a] Andreas Lang and Jana Dittmann. Profiles for evaluation - the usage of audio WET. In Security and Watermarking of Multimedia Contents IX, San Jose, California, U.S.A., 2006. The Society for Imaging Science and Technology (IS&T) and the International Society for Optical Engineering (SPIE), SPIE.


