Theoretical framework for a practical evaluation and comparison of audio watermarking schemes in the triangle of robustness, transparency and capacity

Jana Dittmann[1], David Megías[2], Andreas Lang[1], Jordi Herrera-Joancomartí[2]

[1] Otto-von-Guericke University of Magdeburg, Germany
[2] Universitat Oberta de Catalunya, Spain

Abstract. Digital watermarking is a growing research area to mark digital content (image, audio, video, etc.) by embedding information into the content itself. This technique opens or provides additional and useful features for many application fields (like DRM, annotation, integrity proof and many more). The role of watermarking algorithm evaluation (in a broader sense benchmarking) is to provide a fair and automated analysis of a specific approach if it can fulfill certain application requirements and to perform a comparison with different or similar approaches. Today most algorithm designers use their own methodology and therefore the results are hardly comparable. Derived from the variety of actually presented evaluation procedures in this paper, firstly we introduce a theoretical framework for digital robust watermarking algorithms where we focus on the triangle of robustness, transparency and capacity. The main properties and measuring methods are described. Secondly, a practical environment shows the predefined definition and introduces the practical relevance needed for robust audio watermarking benchmarking. Our goal is to provide a more partial precise methodology to test and compare watermarking algorithms. The hope is that watermarking algorithm designers will use our introduced methodology for testing their algorithms to allow a comparison with existing algorithms more easily. Our work should be seen as a scalable and improvable attempt for a formalization of a benchmarking methodology in the triangle of transparency, capacity and robustness.

1 Introduction

Digital watermarking has been proposed for a variety of applications, including content protection, authentication, integrity, verification, digital rights management and annotation or illustration and many more. A variety of watermarking techniques have been introduced with different promises regarding performance, such as transparency, robustness, capacity, complexity and security. Depending on the watermarking application it is currently not easy to objectively evaluate these performance claims. Watermark evaluation methodologies and tools are therefore important to allow an objective comparison of performance claims and facilitate the development of improved watermarking techniques. The performance of watermarking techniques may also be compared with the specific requirements of applications.

The evaluation process can therefore be very complex and the actual research investigates into evaluation approaches with special attacks for images (see, for example, available tools include Stirmark [35], Optimark [30], Checkmark [5]) or for a specific applications like DRM, see, for example, [2] or so-called profiles, see, for example, [21, 32]. In profile based testing we find basic profiles, extended profiles and application profiles. For example a basic profile is the transparency [21], an extended profile is lossy compression [21] and an application profile is the biometric watermarking profile [37, 22]. For the later beside the evaluation of basic watermark performance parameters, the impact of the introduced signal distortions to the overall biometric error rates for user authentication needs to be evaluated.

In general, a distinction between two types of attackers [24] exists. The first type of attackers uses a single specific or a combination of signal modifications to destroy the watermark or to confuse the detection/retrieval function explicitly. These attackers are malicious and can be classified as powerful. The other type of attackers use the audio signal in a normal environment and they produce single or combined signal modifications implicitly without the goal to destroy the watermark (for example lossy compression). These attackers are non-malicious and they are not interested in attacking the digital audio watermark explicitly. In this paper, we do not distinguish between these types of attackers. A classification of general watermarking attacks to evaluate the robustness is introduced for example in [19]. Therein attacks are classified into removal, geometrical, security and protocol attacks. [38] extends the definition by including estimation attacks or [16, 31, 3, 4] introduce attacks to gain knowledge about the secrets of the system (embedding and/or detection/retrieval) to also evaluate the security of watermarking algorithms. Besides the focus on the robustness and security evaluation for example in [18] the transparency of different steganographic and watermarking algorithms is analyzed.

The goal of our work is to design a theoretical framework to describe and formalize three evaluation properties, namely, robustness, transparency and capacity. With this formalization, watermarking algorithms can be evaluated and their performance can be compared more precisely. Specific measures to evaluate fragility for integrity verification are beyond the scope of our discussion and open for future work. The interested reader find first formalizations for content fragility in [9].

From the software evaluation strategies [1], two general different approaches are known: glass (white) and black box tests. In our case, the watermarking algorithm can be seen as a box, where the cover signal and additional parameters are the input and the marked signal is the output of the box. The two existing types of boxes can be seen as follows. In case of a black box, the evaluation function does not know anything about the watermarking algorithm itself. Therefore, it is unknown for example in which domain the watermark is embedded and what the meaning of the parameters is. The opposite of it is the glass box, where testing involves the knowledge of the watermarking algorithm internals. Hence for example, the working domain and the detailed embedding technique are known. For the evaluation function, it can be very helpful to know the internals about the watermarking algorithm, because the identification of the parameters for a special application field can easily be optimized. Furthermore, if a new watermarking algorithm is introduced the evaluation can make use of the previous knowledge from other algorithms to reduce the costs (time and resources) needed to evaluate it completely. In case of black box testing where only the functional properties are known, the whole evaluation function starts at the beginning with, for example a brute force strategy to optimize the watermarking algorithm parameters. Also a third class of boxes exists, the gray box [1]. In that case, algorithm testing design is educated by information about the watermarking algorithm, like the type of parameters for the program behavior. If, for example, the watermarking embedding function needs a frequency range as a parameter, then its is suggested, that the embedding function works in a specific frequency domain and the evaluation could be tuned for it.

In our work, we set our focus for the practical framework only on black box evaluation, even if the type and usage of the parameters is known. That means that we use brute force mechanisms to evaluate the properties of different embedding algorithms with (if possible) different parameter settings.

The paper is structured as follows. Section 2 introduces the theoretical framework and discusses the properties capacity, transparency and robustness in detail for the embedding, attacking and detection/retrieval evaluation functions. Based on the theoretical framework, our practical evaluation is performed and introduced in section 3. The goal is to show the practicability and applicability of the theoretical framework with five selected audio watermarking algorithms. Section 4 shows and discusses the test results and provides the parameter based comparison within the triangle of robustness, transparency and capacity. The paper closes in section 5 with the conclusions and some suggestions for future work.

2 Theoretical Framework

In this section, the theoretical framework to describe watermarking algorithms and its properties is presented. Furthermore, the formalization of measured test results to provide comparability is introduced. Therefore, we begin with basic definitions, followed by the formalization of selected properties for the evaluation of embedding, detection/retrieval and attacking function. From our formalization, evaluation measures are derived and the interested reader can easily enhance our framework and its introduced methodology.

2.1 Basic Definitions

In this subsection the basic definitions of the theoretical framework 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 $\mathit{\Omega}$ can be defined as the 7-tuple given by

$$\mathit{\Omega} = \left(E, D, R, M, \mathcal{P}_E, \mathcal{P}_D, \mathcal{P}_R\right), \tag{1}$$

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 parameters 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 the 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 [6].

Three important properties of watermarking schemes are usually applied to assess performance, namely *robustness*, *capacity* and *transparency* [12]. 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 1 introduces the triangle between the three properties on two examples [8]. The embedding parameters for the watermarking scheme $\mathit{\Omega}_A$ are tuned to provide high robustness. The price for the robustness of $\mathit{\Omega}_A$ is a bad transparency and a low embedding capacity. Therefore $\mathit{\Omega}_A$ is located close to the robustness corner of the triangle. Watermark $\mathit{\Omega}_B$ is tuned for a high transparency. The result is a low robustness and a low capacity. Therefore $\mathit{\Omega}_B$ is located close to the transparency corner of the triangle.

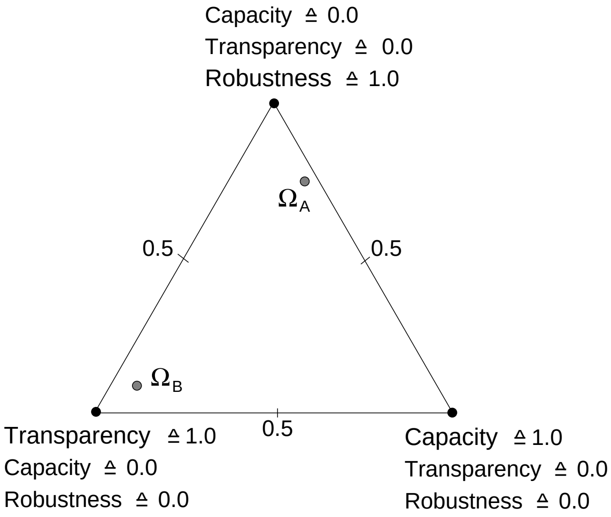

Fig. 1. Illustration of the trade-off between robustness, transparency and capacity

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 1. The requirements of the properties depend on the application used. Remark: unfortunately, two different algorithms, one with 50% transparency, 50% capacity and 50% robustness and the other with 100% transparency, 100% capacity and 100% robustness, would produce the same position middled of the triangle.

Instance of a Watermarking Scheme The Equation (1) defines a general watermarking scheme where several parameters can adopt different values. In particular, there are embedding parameters $\boldsymbol{p}_E \in \mathcal{P}_E$, detection parameters $\boldsymbol{p}_D \in \mathcal{P}_D$ and retrieval parameters $\boldsymbol{p}_R \in \mathcal{P}_R$. Hence, each watermarking scheme Ω may have different instances according to the values that these parameters may adopt. We define an instance Ω^* of the watermarking scheme Ω for a particular value of the parameter vectors:

$$\Omega^* = (E, D, R, M, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) , \tag{2}$$

for $\boldsymbol{\alpha} \in \mathcal{P}_E$, $\boldsymbol{\beta} \in \mathcal{P}_D$ and $\boldsymbol{\gamma} \in \mathcal{P}_R$.

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, digital images, video and any other type of object which can be marked. Once the message is embedded into the object S, a marked object $\hat{S}$ 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 [10]. Hence, we introduce the notation w to denote the *watermark* (or *mark*) which refers to the true embedded bit stream. 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:

$$w = \mathrm{cod}(m, \boldsymbol{p}_{\mathrm{cod}}), \tag{3}$$

where cod is some coding function and $\boldsymbol{p}_{\mathrm{cod}} \in \mathcal{P}_{\mathrm{cod}}$, with $\mathcal{P}_{\mathrm{cod}} \subseteq \mathcal{P}_E$, 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 $\hat{S}$. This kind of schemes are 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 or $M = \{0,1\}^n$ and will be modulated in w. This kind of schemes are usually referred to as *multiple bit watermarking* – or *non zero-bit watermarking schemes*.

2.2 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 $\boldsymbol{p}_E$, the marked object $\hat{S}$ is obtained by means of an embedding function E as follows:

$$\hat{S} = E(S, w, \boldsymbol{p}_E) = E\left(S, \mathrm{cod}(m, \boldsymbol{p}_{\mathrm{cod}}), \boldsymbol{p}_E\right), \tag{4}$$

where specific values must be provided for the coding and the embedding parameters $\boldsymbol{p}_{\mathrm{cod}}$ and $\boldsymbol{p}_E \in \mathcal{P}_E$ and $\mathcal{P}_E$ denotes the domain for the embedding parameters.

The embedding process can usually be tuned with different parameters. Some examples of which kind of parameters can be used are provided in Section 3. 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 $\boldsymbol{p}_E$ of embedding parameters. If a watermarking scheme embeds m multiple times and can be controlled by a parameter $p_{\max}$, then it is part of $\boldsymbol{p}_E$.

Embedding Capacity The embedding capacity 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 cap_E^* would be related to the size of the embedded message, *i.e.* $\mathrm{cap}_E^*(\Omega, \hat{S}) = \mathrm{size}(m) = |m|$. In addition, capacity is often given relative to the size of the cover object:

$$\mathrm{cap}_{E\,\mathrm{rel}}(\Omega^*, \hat{S}) = \frac{\mathrm{cap}_E^*}{\mathrm{size}(\hat{S})}. \tag{5}$$

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 *repeated 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 subsection 2.3.

Embedding Transparency *Transparency (or Imperceptibility) function.* Given a reference object S_{ref} and a test object S_{test} the transparency function T provides a measure of the perceptible distortion between S_{ref} and S_{test}[3]. 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_{ref} and S_{test} are so different that S_{test} cannot be recognized as a version of S_{ref}) and 1 is the best case (an observer does not perceive any significant difference between S_{ref} and S_{test}):

$$T(S_{\mathrm{ref}}, S_{\mathrm{test}}) \to [0, 1]. \tag{6}$$

In case of signal to noise ratio (SNR) measures, the transparency function can be chosen as follows:

$$\overline{\mathrm{SNR}}(S_{\mathrm{ref}}, S_{\mathrm{test}}) = \max(0, 10 \log_{10} \mathrm{SNR}(S_{\mathrm{ref}}, S_{\mathrm{test}})) \tag{7}$$

$$T_{\mathrm{SNR}}(S_{\mathrm{ref}}, S_{\mathrm{test}}) = 1 - \exp(-k \cdot \overline{\mathrm{SNR}}(S_{\mathrm{ref}}, S_{\mathrm{test}})) \tag{8}$$

where k is some positive constant which can be chosen to provide an appropriate scale. Note that for $\mathrm{SNR}(S_{\mathrm{ref}}, S_{\mathrm{test}}) \in (-\infty, 0]$ dB, $T_{\mathrm{SNR}}(S_{\mathrm{ref}}, S_{\mathrm{test}}) = 0$. If we choose $k = 0.075$, then SNR $= 10$ dB implies $T_{\mathrm{SNR}} = 0.52$ and SNR $= 30$ dB implies $T_{\mathrm{SNR}} = 0.89$.

In this paper, however, we have used ODG measures instead of SNR and another transparency function is introduced in Section 3.

We can define a relative transparency for a watermarking scheme Ω^* and a particular object S as follows:

$$\mathrm{tra}_{E\,\mathrm{rel}}(\Omega^*, S) = T(S, \hat{S}), \tag{9}$$

where $\hat{S}$ is obtained as per the embedding function Equation (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

[3] Note that signal to noise ratio (SNR) for audio or peak signal to noise ratio (PSNR) for images is widely used as transparency measure

not related to a particular object S. A definition of “absolute” transparency is related to **a family $\mathcal{S}$ of objects** to be marked and we could apply any of the following definitions:

– Average transparency:

$$\mathrm{tra}_{E\,\mathrm{ave}}(\Omega^*) = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \mathrm{tra}_{E\,\mathrm{rel}}(\Omega^*, S). \tag{10}$$

– Maximum transparency:

$$\mathrm{tra}_{E\,\max}(\Omega^*) = \max_{S \in \mathcal{S}} \left\{ \mathrm{tra}_{E\,\mathrm{rel}}(\Omega^*, S) \right\}. \tag{11}$$

– Minimum transparency:

$$\mathrm{tra}_{E\,\min}(\Omega^*) = \min_{S \in \mathcal{S}} \left\{ \mathrm{tra}_{E\,\mathrm{rel}}(\Omega^*, S) \right\}. \tag{12}$$

2.3 Detection and Retrieval Function

This subsection is devoted to the question related to watermark or message detection and retrieval.

Detection Function Given a test object $\tilde{S}$ (which is suspected to be a possibly attacked or modified version of the marked object $\hat{S}$), a vector of embedding parameters $\boldsymbol{p}_E$, a vector $\boldsymbol{p}_D \in \mathcal{P}_D$ 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:

$$D(\tilde{S}, \boldsymbol{p}_E, \boldsymbol{p}_D, [S, m]) \to \{0, 1\}, \tag{13}$$

where D returns 1 if m is detected in $\tilde{S}$ 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 $\mathcal{K}$ ($k \in \mathcal{K}$) is a component of the vector $\boldsymbol{p}_E$, which is a parameter vector introduced in Equation (13).

Retrieval Function The definition of a retrieval function is only appropriate in non zero-bit watermarking schemes. Given a test object $\tilde{S}$ (which suspected to be a possibly attacked or modified version of the marked object), a vector of embedding parameters $\boldsymbol{p}_E$, a vector $\boldsymbol{p}_R \in \mathcal{P}_R$ of retrieval parameters, the domain $\mathcal{P}_R$ 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:

$$m' = R(\tilde{S}, \boldsymbol{p}_E, \boldsymbol{p}_R, [S, m]), \tag{14}$$

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

In case of repeated 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 $\boldsymbol{p}_R$, the maximum number of multiple embedded m is known and denoted as $p_{\max}$. Furthermore, if the watermarking scheme requires a public or private key for the retrieval process, then the key k belonging to a key space $\mathcal{K}$ ($k \in \mathcal{K}$) is a component of the vector $\boldsymbol{p}_E$, which is a parameter vector introduced in Equation (14).

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 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 1. Verification cases

	Detection	Retrieval
Zero-bit watermarking	w in $\tilde{S}$? (yes/no)	not available
n-bit watermarking	w in $\tilde{S}$? (yes/no)	m'

Classification According to the Information Needed by the Detection or Retrieval Function The 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*.

Retrieval Capacity Now we can define 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 we should consider capacity *after* data extraction. Thus, given the retrieval function of Equation (14), we can define the following capacity $\mathrm{cap}^*_{R\mathrm{rel}}$ function:

$$\mathrm{cap}^*_{R\mathrm{rel}}(\Omega^*, \tilde{S}) = |m| - \sum_{i=1}^{|m|} m_i \oplus m'_i, \tag{15}$$

where $m = m_1 m_2 \dots m_{|m|}$, $m' = m'_1 m'_2 \dots m'_{|m|}$ 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 repeated coding, the retrieved message will be several times longer than the embedded message: $m' = m'_{11} m'_{12} \dots m'_{1|m|} m'_{21} m'_{22} \dots m'_{2|m|} \dots m'_{p_{\max}|m|}$. In such a situation, the retrieval capacity should consider all the repetitions as follows[4]:

$$\mathrm{cap}^{*}_{R\mathrm{rel}}(\Omega^{*}, \tilde{S}) = \sum_{j=1}^{p_{\max}} \left[|m| - \sum_{i=1}^{|m|} m_i \oplus m'_{ji} \right] , \tag{16}$$

where $p_{\max}$ is the counted number of maximal retrieved m'. In the sequel, no repeated 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 $p_{\max} = 1$, which provides Equation (15).

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:

$$\mathrm{cap}_{R\mathrm{rel}}(\Omega^{*}, \tilde{S}) = \frac{\mathrm{cap}^{*}_{R\mathrm{rel}}(\Omega^{*}, \tilde{S})}{\mathrm{size}(\tilde{S})} \tag{17}$$

and it is assumed that the sizes of S, and $\tilde{S}$ are, at least, similar. This second definition provides measures such as bits per pixel (in image watermarking), bits per second (in audio watermarking) 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 $\mathrm{cap}_{R\mathrm{rel}}$ is relative to a given pair $\tilde{S}$ 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 $p_{\max}$. If $p_{\max}$ is unknown, the measure of $\mathrm{cap}^{\dagger}_{R\mathrm{rel}}$ can also be used, but would result in greater (not normalized) values:

$$\mathrm{cap}^{\dagger}_{R\mathrm{rel}}(\Omega^{*}, \tilde{S}) = \frac{\mathrm{cap}^{*}_{R\mathrm{rel}}(\Omega^{*}, \tilde{S})}{|m|\, p_{\max}}. \tag{18}$$

Detection Success Function To measure the overall success of a detection or retrieval function, we introduce a *detection success* function (see Equation (13)). Therefore, the zero-bit a n-bit watermarking scheme are introduced as follows.

For zero-bit watermarking schemes, $\det_D$ returns 0 if the watermark could not be successful detected and 1 if the detection function is able to detect the

[4] 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.

watermark:

$$\mathrm{det}_D(\Omega^*, \tilde{S}) = \begin{cases} 0, \text{no successful detection (negative)}, \\ 1, \text{positive successful detection (positive)}. \end{cases} \tag{19}$$

To measure the successfully embedding rate over a test set, the average of det_D can be computed as follows:

$$\mathrm{det}_{D\mathrm{ave}}(\Omega^*) = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \mathrm{det}_D \tag{20}$$

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 ($p_{\max}$), and the retrieval function $\mathrm{cap}^*_{R\mathrm{rel}}$ 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.

$$\mathrm{det}_R(\Omega^*, \tilde{S}) = \begin{cases} 1, \exists j \in \{1, \dots, p_{\max}\} : \displaystyle\sum_{i=1}^{|m|} m'_{ji} \oplus m_i = 0, \\ 0, \text{otherwise}. \end{cases} \tag{21}$$

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

$$\mathrm{det}_{R\tau}(\Omega^*, \tilde{S}) = \begin{cases} 1, \text{if } \mathrm{cap}^\dagger_{R\mathrm{rel}}(\Omega^*, \tilde{S}) \geq \tau, \\ 0, \text{otherwise}. \end{cases} \tag{22}$$

i.e. detection is reported if the ratio of correctly recovered bits is above some threshold τ (which will be equal to or close to 1).

To measure the successfully embedding rate over a test set, the average of det_R can be computed as follows:

$$\mathrm{det}_{R\mathrm{ave}}(\Omega^*) = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \mathrm{det}_R \tag{23}$$

and the average of $\mathrm{det}_{R\tau}$ as follows:

$$\mathrm{det}_{R\tau\,\mathrm{ave}}(\Omega^*) = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \mathrm{det}_{R\tau} \tag{24}$$

2.4 Attacking Functions

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

$$\tilde{S} = A(\hat{S}, \boldsymbol{p}_A), \tag{25}$$

where $\boldsymbol{p}_A \in \mathcal{P}_A$ is a set of attacking parameters.

Usually, a family of attacking functions $A_{i,j} \in \mathcal{A}$ exists which may be applied to some object, where i identifies an attack and j is a related parameter combination ($\boldsymbol{p}_{A_{i,j}}$). It is assumed that attacks are "simple". If composite attacks $A_{11} \circ A_{12} \circ \cdots \circ A_{1j} \circ A_{21} \circ \cdots \circ A_{2j} \circ \cdots \circ A_{i_{\max} j_{\max}}(\hat{S})$ 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 [20, 24, 23, 21, 39].

Robustness A watermarking scheme Ω 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. Ω is defined as fragile if the detection function D or retrieval function R is not able to recover the mark *i.e.*to detect after a malicious attack signal change. In our discussion, we conclude on the robustness not on the fragility for detection of content changes. Here specific enhancements like the definition of malicious and non-malicious changes (attacks) become important and are out of the scope of the paper. However, the definition of robustness only classifieswatermarking 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 $A_{i,j}(\hat{S}) = \varnothing$, where $\varnothing$ 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: $T(\tilde{S}, \varnothing) \approx 0$. 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 which destroys the embedded mark and, at the same time, produces little distortion exists, 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 $\mathrm{rob}_{\mathrm{rel}}$ 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 is used to measure the robustness. If the robustness is measured based on the byte error rate $\mathrm{rob}^{\mathrm{byte}}$, then a given watermarking scheme is classified as robust if the bytes of the embedded massage (characters) are correctly retrieved. Another robustness measure function based on the bit error rate $\mathrm{rob}^{\mathrm{bit}}$ returns the percentage robustness of the watermarking scheme measured over the whole attacking and test set. For zero-bit watermarking schemes no retrieval function exists and no classification based on bit or byte error rates is possible. To simplify matters, the robustness measure for zero-bit watermarking schemes is always classified to $\mathrm{rob}^{\mathrm{byte}}$. 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 $3 \times 8 = 24$ 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 and one is false, which has a value of $\frac{1}{3} = 0.3\overline{3}$. The bit error rate returns, that 18 bits are correct and 6 bits are false, which has a value of $\frac{6}{24} = 0.25$. If the 1st, 2nd, 8th, 9th, 16th and 17th bits are destroyed, then the byte error rate returns that all bytes (characters) are false and the result has a value of $\frac{3}{3} = 1.0$ which means that 100% 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, differences are apparent in the byte error rater. Therefore, the following equations introduce the robustness for n-bit watermarking schemes divided into $\mathrm{rob}^{\mathrm{byte}}$ and $\mathrm{rob}^{\mathrm{bit}}$ and for zero-bit watermarking schemes only for $\mathrm{rob}^{\mathrm{byte}}$. The two robustness measures $\mathrm{rob}^{\mathrm{byte}}$ and $\mathrm{rob}^{\mathrm{bit}}$ returns completely different robustness values. We introduce them to show that different approaches are possible and depending on test goals, choices are to be made to select the measure function. We note, that different measure methods are available to measure the robustness, *i.e.*based on $\det_R$ 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 $\tilde{S} = A_{i,j}(\hat{S})$:

$$\mathrm{rob}_{\mathrm{rel}}^{\mathrm{byte}}(\Omega^*, \hat{S}) = 1 - \max_{A_{i,j} \in \mathcal{A}} \left\{ T\left(\hat{S}, \tilde{S}\right) : \det\nolimits_D \left(\tilde{S}, \boldsymbol{p}_E^{\mathrm{opt}}, \boldsymbol{p}_D^{\mathrm{opt}}, \boldsymbol{p}_{\mathrm{cod}}, [S, m]\right) = 0 \right\}, \tag{26}$$

and for a n-bit watermarking scheme:

$$\mathrm{rob}_{\mathrm{rel}}^{\mathrm{byte}}(\Omega^*, \hat{S}) = 1 - \max_{A_{i,j} \in \mathcal{A}} \left\{ T\left(\hat{S}, \tilde{S}\right) : \det\nolimits_R \left(\tilde{S}, \boldsymbol{p}_E^{\mathrm{opt}}, \boldsymbol{p}_D^{\mathrm{opt}}, \boldsymbol{p}_{\mathrm{cod}}, [S, m]\right) = 0 \right\}, \tag{27}$$

that is, given a marked object $\hat{S}$ and all the attacks which destroy the mark, even for optimal embedding and detection parameters (p_E^{opt}, p_D^{opt}), the one which produces less distortion in the marked object $\hat{S}$ determines how robust the scheme is. If none of the attacks in the family $\mathcal{A}$ erases the embedded mark, then this measure is (by definition) equal to 1 (the best possible value).

The functions provided in Equation (26), Equation (27) and Equation (31) 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 (27) 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 Ω 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 (27) is 0.77.

However, the functions of Equation (26) and Equation (27) are *relative* to a given object $\tilde{S}$ (hence the use of the subindex ”rel”) but we usually want 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 ($\mathrm{rob}_{\mathrm{rel}}^{\mathrm{byte}}$) which can be defined in several ways. Given a family $\mathcal{S}$ of cover objects, and their corresponding marked objects $\hat{S}$ obtained by means of the embedding equation Equation (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:

$$\mathrm{rob}_{\mathrm{ave}}^{\mathrm{byte}}(\Omega^*) = \frac{1}{|\mathcal{S}|}\sum_{S\in\mathcal{S}} \mathrm{rob}_{\mathrm{rel}}^{\mathrm{byte}}(\Omega^*, \hat{S}). \tag{28}$$

- Minimum robustness (worst case approach) based on byte error rate:

$$\mathrm{rob}_{\mathrm{min}}^{\mathrm{byte}}(\Omega^*) = \min_{S\in\mathcal{S}} \mathrm{rob}_{\mathrm{rel}}^{\mathrm{byte}}(\Omega^*, \hat{S}). \tag{29}$$

- Probabilistic approach based on byte error rate:

$$\mathrm{rob}_{\mathrm{prob}}^{\mathrm{byte}}(\Omega^*, r) = 1 - \underset{S\in\mathcal{S}}{p}\left(\mathrm{rob}_{\mathrm{rel}}^{\mathrm{byte}}(\Omega^*, \hat{S}) < r\right), \tag{30}$$

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

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.

Another robustness measure based on the bit error rate related to the transparency for n-bit watermarking schemes can be defined as:

$$\mathrm{rob}_{\mathrm{ave}}^{\mathrm{bit}}(\Omega^*) = \frac{1}{|\mathcal{S}||\mathcal{A}|}\sum_{S\in\mathcal{S}}\sum_{A_{i,j}\in\mathcal{A}} \begin{cases} 0, \left(\mathrm{cap}_{R\mathrm{rel}}^{\dagger} < \tau\right) \wedge (\mathrm{tra}_{A\mathrm{rel}} > \nu) \\ 1, \text{otherwise} \end{cases}, \tag{31}$$

Attacking Transparency We can define a relative transparency for the attacking process for a watermarking scheme Ω^* and a particular object S as follows. Two different measures can be provided. The first is the transparency of the attacked object with respect to the marked object (which is the most obvious one):

$$\mathrm{tra}_{A\mathrm{rel}}(\Omega^*, \hat{S}, \tilde{S}) = T(\hat{S}, \tilde{S}), \tag{32}$$

where $\hat{S}$ is obtained as per the embedding function Equation (4) and $\tilde{S} = A_{i,j}(\hat{S})$, $p_{A_{i,j}}$ 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:

$$\mathrm{tra}_{A\mathrm{rel}}^{*}(\Omega^*, S, \tilde{S}) = T(S, \tilde{S}). \tag{33}$$

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 $\hat{S}$. 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. We could thus apply any of the following definitions:

- Average transparency:

$$\mathrm{tra}_{A\mathrm{ave}}(\Omega^*) = \frac{1}{|\mathcal{S}|\,|\mathcal{A}|} \sum_{S\in\mathcal{S}} \sum_{A_{i,j}\in\mathcal{A}} \mathrm{tra}_{A\mathrm{rel}}(\Omega^*, \hat{S}, \tilde{S}). \tag{34}$$

- Maximum transparency:

$$\mathrm{tra}_{A\mathrm{max}}(\Omega^*) = \max_{S\in\mathcal{S}} \left\{ \max_{A_{i,j}\in\mathcal{A}} \left\{ \mathrm{tra}_{A\mathrm{rel}}(\Omega^*, \hat{S}, \tilde{S}) \right\} \right\}. \tag{35}$$

- Minimum transparency:

$$\mathrm{tra}_{A\mathrm{min}}(\Omega^*) = \min_{S\in\mathcal{S}} \left\{ \min_{A_{i,j}\in\mathcal{A}} \left\{ \mathrm{tra}_{A\mathrm{rel}}(\Omega^*, \hat{S}, \tilde{S}) \right\} \right\}. \tag{36}$$

Note that similar definitions can be provided with respect to $\mathrm{tra}^*_{A\mathrm{rel}}(\Omega^*, S, \tilde{S})$.

Attacking Capacity Finally, the capacity of a watermarking scheme can now be related to a family of attacks $\mathcal{A}$ and a family of objects $\mathcal{S}$ as follows:

- Average capacity:

$$\mathrm{cap}_{A\mathrm{ave}}(\Omega^*) = \frac{1}{|\mathcal{S}|\,|\mathcal{A}|} \sum_{S\in\mathcal{S}} \sum_{A_{i,j}\in\mathcal{A}} \mathrm{cap}_{R\mathrm{rel}}(\Omega^*, \tilde{S}). \tag{37}$$

- Maximum capacity:

$$\mathrm{cap}_{A\mathrm{max}}(\Omega^*) = \max_{S\in\mathcal{S}} \left\{ \max_{A_{i,j}\in\mathcal{A}} \left\{ \mathrm{cap}_{R\mathrm{rel}}(\Omega^*, \tilde{S}) \right\} \right\}. \tag{38}$$

- Minimum capacity:

$$\mathrm{cap}_{A\mathrm{min}}(\Omega^*) = \min_{S\in\mathcal{S}} \left\{ \min_{A_{i,j}\in\mathcal{A}} \left\{ \mathrm{cap}_{R\mathrm{rel}}(\Omega^*, \tilde{S}) \right\} \right\}. \tag{39}$$

Therefore, based on the retrieved capacity $\mathrm{cap}_{R\mathrm{rel}}$ from R, the attacking capacities $\mathrm{cap}_{A\mathrm{ave}}, \mathrm{cap}_{A\mathrm{max}}$ and $\mathrm{cap}_{A\mathrm{min}}$ are introduced as shown above. It is also possible to describe other measurable attacking capacities based on the other two defined retrieving capacities $\mathrm{cap}^*_{R\mathrm{rel}}$ or $\mathrm{cap}^\dagger_{R\mathrm{rel}}$.

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 det_R. Following that successful detection after attacking det_A for a specific attack or $\mathrm{det}_{A\mathrm{ave}}$ for an average value over a set of attacks with $\boldsymbol{p}_A$ can be described for zero-bit watermarking schemes as:

$$\mathrm{det}_A = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \mathrm{det}_D, \text{ for a specific attack } A_{i,j} \tag{40}$$

and for n-bit watermarking schemes as:

$$\mathrm{det}_A = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \mathrm{det}_R, \text{ for a specific attack } A_{i,j} \tag{41}$$

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

$$\mathrm{det}_{A\mathrm{ave}} = \frac{1}{|\mathcal{S}|\,|\mathcal{A}|} \sum_{S \in \mathcal{S}} \sum_{A_{i,j} \in \mathcal{A}} \mathrm{det}_D \tag{42}$$

and for n-bit watermarking schemes as:

$$\mathrm{det}_{A\mathrm{ave}} = \frac{1}{|\mathcal{S}|\,|\mathcal{A}|} \sum_{S \in \mathcal{S}} \sum_{A_{i,j} \in \mathcal{A}} \mathrm{det}_R \tag{43}$$

With $\mathrm{det}_{A\mathrm{ave}}$ the normalized successful detection after attacking can be measured and the result is in the range $[0, 1]$. If the function $\mathrm{det}_{R\mathcal{T}}$ is used to measure the successful detection, then the detection success after attacking would be $\mathrm{det}_{A\mathcal{T}}$ for a specific attack or $\mathrm{det}_{A\mathcal{T}\mathrm{ave}}$ as an average value over a given test set and attacking set.

2.5 Evaluation Methodology for the Triangle

From the introduced parameters and measures in the previous subsections we can now derive an evaluation methodology to analyze one given algorithm (intra–algorithm evaluation or analysis) and to compare different algorithms (inter–algorithm evaluation or analysis) in the triangle created by the embedding, detection/retrieval and attacking function. Our evaluation methodology uses all the defined parameters and measures are summarized in Table 2. 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 transparency, robustness and capacity 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 in the triangle 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 2. Summarizing of evaluation methodology

	embedding	detection/retrieval	attacking
$\Omega_i(E, D, R, M, \mathcal{P}_E, \mathcal{P}_D, \mathcal{P}_R)$	$\boldsymbol{p}_E \in \mathcal{P}_E$,	$\boldsymbol{p}_D \in \mathcal{P}_D, \boldsymbol{p}_R \in \mathcal{P}_R$,	$\boldsymbol{p}_A \in \mathcal{P}_A$
m, m'	cap^*_E, $\mathrm{cap}_{E\mathrm{rel}}$	$\mathrm{cap}^*_{R\mathrm{rel}}$, $\mathrm{cap}_{R\mathrm{rel}}$, $\mathrm{cap}^\dagger_{R\mathrm{rel}}$, det_R, $\mathrm{det}_{R\tau}$, $\mathrm{det}_{R\mathrm{ave}}$	$\mathrm{cap}_{A\mathrm{rel}}$, $\mathrm{cap}_{A\mathrm{ave}}$, $\mathrm{cap}_{A\mathrm{min}}$, $\mathrm{cap}^\dagger_{A\mathrm{ave}}$, $\mathrm{cap}^*_{A\mathrm{ave}}$ $\mathrm{cap}_{A\mathrm{max}}$, det_A, $\mathrm{det}_{A\mathrm{ave}}$, $\mathrm{det}_{A\tau}$, $\mathrm{det}_{A\tau\mathrm{ave}}$, $\mathrm{det}_{A\tau\mathrm{max}}$, $\mathrm{det}_{A\tau\mathrm{min}}$
w		det_D, $\mathrm{det}_{D\mathrm{ave}}$	det_A, $\mathrm{det}_{A\mathrm{ave}}$
S, $\hat{S}$	$\mathrm{tra}_{E\mathrm{rel}}$, $\mathrm{tra}_{E\mathrm{ave}}$, $\mathrm{tra}_{E\mathrm{min}}$, $\mathrm{tra}_{E\mathrm{max}}$		
$\hat{S}$, $\tilde{S}$			$\mathrm{tra}_{A\mathrm{rel}}$, $\mathrm{tra}_{A\mathrm{ave}}$, $\mathrm{tra}_{A\mathrm{min}}$, $\mathrm{tra}_{A\mathrm{max}}$
S,$\tilde{S}$			$\mathrm{tra}^*_{A\mathrm{rel}}$, $\mathrm{tra}^*_{A\mathrm{ave}}$, $\mathrm{tra}^*_{A\mathrm{min}}$, $\mathrm{tra}^*_{A\mathrm{max}}$
$m', \tilde{S}$			$\mathrm{rob}^{\mathrm{byte}}_{\mathrm{rel}}$, $\mathrm{rob}^{\mathrm{byte}}_{\mathrm{ave}}$, $\mathrm{rob}^{\mathrm{byte}}_{\mathrm{min}}$, $\mathrm{rob}^{\mathrm{byte}}_{\mathrm{prob}}$, $\mathrm{rob}^{cod}_{\mathrm{ave}}$

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

Secondly our 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 and placed in one triangle to compare the different performance results from these different parameter setting (intra–algorithm analysis). In our tests, for example, we compare the influence of different attack parameter settings to one specific embedding and detection/retrieval setting to one algorithm. Furthermore, if we compare different algorithms we can place the algorithms in the same or in a different triangle 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, cap^*_E defines the absolute length of m and $\mathrm{cap}^*_{E\mathrm{rel}}$ the relative length of m normalized to the length of the audio signal. For retrieval, $\mathrm{cap}^*_{R\mathrm{rel}}$ defines the absolute lengths of retrieved m'. Therefore, it is used to measure for example the bit error rate (BER) or byte error rate over the whole audio signal. A repeated embedding of m can be identified as well in the $\mathrm{cap}^\dagger_{R\mathrm{rel}}$. The retrieved capacity can be normalized to the length of the audio signal (or frames of it) with $\mathrm{cap}_{R\mathrm{rel}}$ or to the length of m with $\mathrm{cap}^\dagger_{R\mathrm{rel}}$. For attacking, the capacity $\mathrm{cap}_{A\mathrm{ave}}$ defines the normalized average capacity after one or more attacks to an audio test set. Furthermore, $\mathrm{cap}_{A\mathrm{min}}$ and $\mathrm{cap}_{A\mathrm{max}}$ defines the minimum and maximum received capacity after one or more attacks on a given audio test set. The function det_D for a zero-bit watermarking scheme and det_R for n-bit watermarking scheme determine, if a given m can be embedded into an audio signal or not. Therefore, the average values of them $\mathrm{det}_{D\mathrm{ave}}$ and $\mathrm{det}_{R\mathrm{ave}}$ shows the average success of the embedding function by using directly after embedding the detection or retrieval function as verification.

The transparency of the embedding function (between $S, \hat{S}$) can be measured with $\mathrm{tra}_{E\mathrm{rel}}$ for a specific watermarking algorithm and a specific audio signal with a given parameter set. Furthermore, $\mathrm{tra}_{E\mathrm{ave}}$, $\mathrm{tra}_{E\mathrm{min}}$ and $\mathrm{tra}_{E\mathrm{max}}$ 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 $(\hat{S}, \tilde{S})$ is similar measured to the embedding transparency. Therefore, relative ($\mathrm{tra}_{A\mathrm{rel}}$), average ($\mathrm{tra}_{A\mathrm{ave}}$), minimal ($\mathrm{tra}_{A\mathrm{min}}$) and maximal ($\mathrm{tra}_{A\mathrm{max}}$) transparency can be measured and compared. If the attacking transparency is measured between the attacked and original signal $(S, \tilde{S})$, then the same types of transparencies are defined: relative ($\mathrm{tra}^*_{A\mathrm{rel}}$), average ($\mathrm{tra}^*_{A\mathrm{ave}}$), minimal ($\mathrm{tra}^*_{A\mathrm{min}}$) and maximal ($\mathrm{tra}^*_{A\mathrm{max}}$). 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 range $[0, 1]$.

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 $\left(\text{rob}_{\text{ave}}^{\text{byte}}, \text{rob}_{\text{ave}}^{\text{bit}}\right)$, the minimum $\left(\text{rob}_{\text{min}}^{\text{byte}}\right)$ which includes the best attacking transparency and the best detection/retrieving results and a probabilistic result $\left(\text{rob}_{\text{prob}}^{\text{byte}}\right)$. Therefore, m' is retrieved with function R of Ω^* and m must be known to measure $\text{cap}_{R\text{rel}}^{\dagger}$. For $\text{rob}_{\text{ave}}^{\text{bit}}$, the thresholds τ and ν define with the function $\det_{R\tau}$, if Ω^* is robust or not against $A_{i,j}$ by using a detection of m' depending on τ. Furthermore, the results of $\det_{A\tau}$ for a specific or $\det_{A\tau\text{ave}}$ 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 introduced methodology allows intra and inter–algorithm evaluation or analysis as well as the separate selection of embedding, detection/retrieval parameters for Ω^*, the attacking functions and its parameters, the test set $\mathcal{S}$ and the overall attack set $\mathcal{A}$.

3 Practical Evaluation

In this section, we set up a practical evaluation based on the described theoretical framework in order to show how to perform a practical evaluation (comparison) of audio watermarking algorithms. In the subsection 3.1 the five example audio watermarking algorithms and their parameters ($\boldsymbol{p}_E$ and $\boldsymbol{p}_D, \boldsymbol{p}_R$) are introduced. The audio test set and the test scenario (practical evaluation framework) used in subsection 3.2 is shown with attacks $A_{i,j}$ and $\boldsymbol{p}_{Ai,j}$. Our test goals are introduced in subsection 3.3.

From the methodology introduced in section 2.5 we select a subset of measures to perform a proof of concept evaluation (practical usage) of the theoretical framework. Therefore, the following Table 3 shows the prototypical implemented functions.

The methodology allows us to provide an intra–algorithm evaluation and analysis by using different parameter settings for the attacks as well as an inter–algorithm analysis to provide comparability between selected watermarking algorithms.

3.1 Evaluated Watermarking Algorithms: Basic Definitions

For our exemplary evaluation we use five different audio watermarking algorithms ($\Omega_1, \ldots, \Omega_5$). The following description 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. In our later test setup the watermarking algorithms are seen as black boxes.

Table 3. Practical used evaluation

	embedding	detection/retrieval	attacking
$\Omega_i(E, D, R, M, \mathcal{P}_E, \mathcal{P}_D, \mathcal{P}_R)$	$\boldsymbol{p}_E \in \mathcal{P}_E$,	$\boldsymbol{p}_D \in \mathcal{P}_D, \boldsymbol{p}_R \in \mathcal{P}_R$,	$\boldsymbol{p}_A \in \mathcal{P}_A$
m, m'	cap_E^*	$\mathrm{det}_{R\,\mathrm{ave}}$, $\mathrm{det}_{R\tau}$, $\mathrm{cap}_{R\,\mathrm{rel}}^*$	$\mathrm{cap}_{A\,\mathrm{ave}}$, $\mathrm{cap}_{A\,\mathrm{min}}$, $\mathrm{cap}_{A\,\mathrm{max}}$, $\mathrm{cap}_{A\,\mathrm{ave}}^\dagger$, $\mathrm{cap}_{A\,\mathrm{ave}}^\dagger$, $\mathrm{cap}_{A\,\mathrm{ave}}^\dagger$, $\mathrm{det}_{A\,\mathrm{ave}}$, $\mathrm{det}_{A\tau\,\mathrm{ave}}$,
w		$\mathrm{det}_{D\,\mathrm{ave}}$	$\mathrm{det}_{A\,\mathrm{ave}}$
S, $\hat{S}$	$\mathrm{tra}_{E\,\mathrm{ave}}$, $\mathrm{tra}_{E\,\mathrm{min}}$, $\mathrm{tra}_{E\,\mathrm{max}}$		
$\tilde{S}$, $\hat{S}$			$\mathrm{tra}_{A\,\mathrm{ave}}$, $\mathrm{tra}_{A\,\mathrm{min}}$, $\mathrm{tra}_{A\,\mathrm{max}}$
$m', \tilde{S}$			$\mathrm{rob}_{\mathrm{ave}}^{\mathrm{byte}}$

$\boldsymbol{\Omega_1}$: This 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 [33]. 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) [13]. By using Ω_1, the following parameters are defined for this algorithm:

- p_1: specifies the internal embedding method and at present only ZT (zerotree) is possible.
- p_2: specifies the internal coding method and at present, only binary (BIN) is possible. As $\boldsymbol{p}_{\mathrm{cod}} \in \boldsymbol{p}_E$ the coding method used for $\boldsymbol{p}_{\mathrm{cod}}$ is seen as $\boldsymbol{p}_E$.

Embedding Function: As input audio signal S, this watermarking scheme reads only uncompressed PCM audio files in WAVE format. The output signal $\hat{S}$ is only writable in uncompressed PCM WAVE file format. The parameters needed for E are $\boldsymbol{p}_E = (p_1, p_2)$.

Detection/Retrieval Function: As input audio signal $\hat{S}$ or $\tilde{S}$ only uncompressed PCM audio files in WAVE format are supported. Furthermore, there is no distinction between D and R. Therefore, only the retrieval function R can be used. The parameters needed for R are $\boldsymbol{p}_R = (p_1, p_2)$, $D = \varnothing$.

The introduced parameters are subsequently assigned to $\boldsymbol{p}_E$, $\boldsymbol{p}_D$ and $\boldsymbol{p}_R$. Therefore, this watermarking scheme can be described as follows:

$$\Omega_1 = (E, \varnothing, R, m, \{p_1 = BIN, p_2 = ZT\}, \varnothing, \{p_1 = BIN, p_2 = ZT\})$$

Other parameter combinations are currently not available. The working domain of this algorithm is wavelet and can exemplary be described as:

$$\Omega_1 = (E_{wavlet}, \varnothing, R_{wavelet}, m, \{p_1 = BIN, p_2 = ZT\}, \\ \varnothing, \{p_1 = BIN, p_2 = ZT\})$$

Ω_2: This n-bit stream watermarking algorithm works in the frequency domain and embeds the watermark in the frequency coefficients by using a spread spectrum technique [17]. It does not use a secret key and can therefore also be categorized as annotation watermarking scheme. This algorithm has only the message m as input parameter:

- m: specifies the watermarking message, which will be embedded. It can be a string with characters [a-zA-Z0-9].

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 $\hat{S}$ can also be written in uncompressed PCM WAVE file format. There are no parameters for E defined; ($\boldsymbol{p}_E = (\varnothing)$).

Detection/Retrieval Function: As input audio signals $\hat{S}$ or $\tilde{S}$ 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 $\boldsymbol{p}_R = (\varnothing)$.

Therefore, Ω_2 has no parameters for $\boldsymbol{p}_E$, $\boldsymbol{p}_D$ and $\boldsymbol{p}_R$, which can be changed for the embedding and detecting/retrieval function. For an intra–algorithm analysis, only the test set, attack set and/or attacking parameters can be changed.

This watermarking algorithms can be described as follows:

$$\Omega_2 = (E, \varnothing, R, m, \varnothing, \varnothing, \varnothing)$$

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

$$\Omega_2 = (E_{freq}, \varnothing, R_{freq}, m, \varnothing, \varnothing, \varnothing)$$

Ω_3: This n-bit stream watermarking algorithm works in the frequency domain and embeds w ($w = cod(m, \boldsymbol{p}_{\text{cod}})$) 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:

- k: defines the secret key and is an integer value
- p_1: is the scaling factor used to define the embedding strength
- p_2: defines the lower frequency bound in range $[0, \frac{samplerate}{2}]$

- p_3: defines the upper frequency bound in range $[0, \frac{samplerate}{2}]$ and $p_2 \leq p_3$
- p_4: defines the frame size used for the windowing function typical power of 2
- p_5: 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 [26]. The parameters needed for E are $\boldsymbol{p}_E = (k, p_1, p_2, p_3, p_4)$.

Detection/Retrieval Function: Supported input audio signals $\hat{S}$ or $\tilde{S}$ are all file formats provided by the *libsndfile* library. The implementation of Ω_3 does not distinguish between D and R. Therefore, only the retrieval function R can be used. The parameters needed for R are $\boldsymbol{p}_R = (k, p_2, p_3, p_4, p_5)$.

The maximum frequency of the frequency bound depends on the sampling rate and is defines as $f_{tot} = \frac{sampling\ rate}{2}$ [15]. Ω_3 can be described as follows:

$$\Omega_3 = (E, \varnothing, R, m, \{k, p_1 \in [0, \infty], p_2 \in [0, f_{tot}], p_3 \in [0, f_{tot}], \\ p_4 = 2^x, x \in N\}, \varnothing, \{p_2 \in [0, f_{tot}], p_3 \in [0, f_{tot}], t \in [0, 1], p_4, p_5\})$$

The constrain $p_2 \leq p_3$ needs to be satisfied. The working domain of this algorithm is also the frequency domain and can be described as:

$$\Omega_3 = (E_{freq}, \varnothing, R_{freq}, m, p_1 \in [0, \infty], \{p_2 \in [0, f_{tot}], p_3 \in [0, f_{tot}], \\ p_4 = 2^x, x \in N\}, \varnothing, \{p_2 \in [0, f_{tot}], p_3 \in [0, f_{tot}], t \in [0, 1], p_4, p_5\})$$

Ω_4: This watermarking algorithm is classified as a zero-bit watermark. It works in the wavelet domain and embeds the watermark in selected coefficients [11]. To embed the watermark into the audio signal a three level DWT domain and a Daubechies 8-tap filter is used [11]. The following parameters can be defined:

- k: defines the secret key as integer value
- p_1: defines a threshold, which selects the coefficients for embedding. The default value is $p_1 = 40$
- p_2: defines a scale factor and which describes the embedding strength. The default value is $p_2 = 0.2$.

Embedding Function: As input audio signal S, this watermarking scheme reads and writes all file formats provided by the *libsndfile* library [26]. The parameters needed for E are $\boldsymbol{p}_E = (k, p_1, p_2)$.

Detection/Retrieval Function: Supported input audio signals $\hat{S}$ or $\tilde{S}$ are all file formats provided by the *libsndfile* library. Only the detection is possible and the parameters for D are $\boldsymbol{p}_D = (k, p_1, p_2)$.

Therefore, Ω_4 is a zero-bit watermarking scheme, only D can be used for detection.

This watermarking algorithms can be described as follows:

$$\Omega_4 = (E, D, \varnothing, \varnothing, \boldsymbol{p}_E(k, t, s), \boldsymbol{p}_D(k, t, s), \varnothing)$$

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

$$\Omega_4 = (E_{wavelet}, D_{wavelet}, \varnothing, \varnothing, \boldsymbol{p}_E(k,t,s), \boldsymbol{p}_D(k,t,s), \varnothing)$$

$\boldsymbol{\Omega_5}$: This watermarking algorithm [27, 28] works in frequency domain and embeds the watermark w at different frequencies which are chosen by comparing the original audio signal S with a modified version S' which is obtained using an MP3 compressor and decompressor multiple times. The watermark w is built using a Dual Hamming code $DH(31,5)$ for error correction from the message m ($w = cod(m,31,5) = DH(m,31,5)$) and repeated coding is used. Both detection and retrieval functions are implemented. The retrieval function recovers all the repetitions of the message m and the detection one uses a voting scheme to determine a single value for each message bit. Then, the identified value is compared to the original and detection is reported if 90% or more bits are correctly recovered. In addition, a secret key k is used to generate a pseudo-random sequence which is added to the watermark prior to embedding in order to generate a non-repetitive binary sequence in the embedding process. This step is intended to avoid some types of attacks which may exploit the cyclic repetition of the same bits at different frequencies. In summary:

- m is a n-bit stream which defines the transmitted message.
- w is the message m encoded using the $w = cod(m, DH(31,5))$.
- k: secret key, as 64 bit long value, therefore $k \in [0, 2^{64} - 1]$

Embedding function: The following embedding parameters $\boldsymbol{p}_E$ are used:

- p_1: bit rate of the MP3 compressor/decompressor.
- $p_2 \in [0, 100]$: percentage of the maximum magnitude to choose the relevant frequencies.
- $p_3 \in [0, 1]$: maximum relative error between the magnitudes of the original and modified (compressed-decompressed) signals to choose a frequency.
- $p_4 \in [0, \infty)$: magnitude modification parameter (in dB).

Retrieval function: The following embedding parameters $\boldsymbol{p}_D$ are used:

- $q \in [0, 100]$: percentage (tolerance) to recover the embedded bits.

Detection function: the embedded message is identified and a voting scheme is applied to obtain a single copy of each bit. If 90% or more bits are identical, detection is returned. Thus, detection is built in terms of retrieval.

This watermarking algorithms can be described as follows:

$$\Omega_5 = (E, \varnothing, R, m, (p_1 = 128, p_2 = 5, p_3 = 0.02, p_4 = 0.2), \varnothing, q = 2)$$

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

$$\Omega_5 = (E_{freq}, \varnothing, R_{freq}, m, (p_1 = 128, p_2 = 5, p_3 = 0.02, p_4 = 0.2), \varnothing, q = 2)$$

The different parameters for values for $\Omega_1, \ldots, \Omega_5$ have been chosen according to the tuning guidelines provided in [29] and are summarized in the following Tables 4 and 5.

Table 4. Used embedding parameters $\boldsymbol{p}_E$

Algorithm	embedding parameters
Ω_1^*	$\boldsymbol{p}_E = (p_1 = BIN, p_2 = ZT)$
Ω_2^*	$\boldsymbol{p}_E = (\varnothing)$
Ω_3^*	$\boldsymbol{p}_E = (k = 1234, p_2 = \{500, 2000\}, p_3 = \{5000, 10000\}, p_1 = \{1.5, 3\}, p_4)$
Ω_4^*	$\boldsymbol{p}_E = (k = 1234, p_1 = 0.05, p_2 = 40)$
Ω_5^*	$\boldsymbol{p}_E = [k, p_1 = 128\ \text{kbps}, p_2 = 5, p_3 = 0.02, p_4 = 0.2\ \text{dB}]^{\mathrm{T}}$

where the superscript "T" denotes the transposition operation and the key used for Ω_5^* is $k = \text{A71CD57159DA9E2D}_{(16)}$. The footnote $_{(16)}$ indicates the key space.

Table 5. Used detection/retrieval parameters $\boldsymbol{p}_D$ and $\boldsymbol{p}_R$

Algorithm	detection/retrieval parameters
Ω_1^*	$\boldsymbol{p}_R = (p_1 = BIN, p_2 = ZT)$
Ω_2^*	$\boldsymbol{p}_R = (\varnothing)$
Ω_3^*	$\boldsymbol{p}_R = (k = 1234, p_2 = \{500, 2000\}, p_3 = \{5000, 10000\}, p_5 = 0.6, p_4)$
Ω_4^*	$\boldsymbol{p}_D = (k = 1234, p_1 = 0.05, p_2 = 40)$
Ω_5^*	$\boldsymbol{p}_R = \boldsymbol{p}_D = (k = \text{A71CD57159DA9E2D}_{(16)}, q = 2)$

To show a practical test setup, we choose the following methodology.

If the watermarking algorithm is a zero-bit-watermark, then the *result* of $\det_D$ can be a 1 (*yes*) if the watermark is present in $\tilde{S}$ or 0 (*no*) if it is not detectable in $\tilde{S}$ depending on $\boldsymbol{p}_D$. Otherwise, if the watermarking algorithms is a n-bit-watermark, the $w = cod(m, \boldsymbol{p}_{\text{cod}})$ is computed for E and R retrieves m' from $\tilde{S}$ with its parameters $\boldsymbol{p}_R$. For both types of algorithms, the robustness (rob_{rel}) and the transparency ($\text{tra}_{A\text{ave}}$) of $A_{i,j}$ are measured. The Table 6 shows the used watermarking algorithms and its type of classification and if the watermarking algorithms are categorized as a secure scheme (key needed) or not.

Table 6. Types of evaluated watermarking algorithms

watermarking algorithm	type of classification	key required
Ω_1	n-bit watermark	no
Ω_2	n-bit watermark	no
Ω_3	n-bit watermark	yes
Ω_4	zero-bit watermark	yes
Ω_5	n-bit watermark	yes

3.2 Test Scenario – The Practical Framework

In this subsection the used audio test set is introduced and the audio signals and its characteristics are being described. Furthermore, the test set as well as the attacking functions are introduced and summarized.

All five watermarking algorithms use the same audio test set S_{SQAM} which contains 16 different uncompressed audio files for $S \in S_{SQAM}$. The audio signals are the well known SQAM files [34]. All audio signals are in CD quality and they have a sampling rate of $44.1kHz$ with two audio channels (stereo) and $16bit$ sample resolution. The minimal length of an audio signal is $16.3s$, the maximum length $34.9s$ and the average length of all audio signals $21.26s$. Furthermore, the audio files are categorized in three types of content, which is shown in Table 7. Therefore, the first category *single instrument* contains 7 audio files, where a single music instrument is audible, the second category *speech* contains spoken text with female and male voices in the languages English, German and French. The last category *singing* contains female, male and a mixture of both singing voices.

Table 7. Audio files and its classification used for the test scenario

single instruments	speech	singing
harp40_1.wav	spfe49_1.wav	bass47_1.wav
horn23_2.wav	spff51_1.wav	sopr44_1.wav
trpt21_2.wav	spfg53_1.wav	quar48_1.wav
vioo10_2.wav	spme50_1.wav	
gspi35_1.wav	spmf52_1.wav	
gspi35_2.wav	spmg54_1.wav	
frer07_1.wav		

Our test scenario is as follows. All audio signals S are used as cover medium. The embedding function E and its selected parameters $\boldsymbol{p}_E$ embeds the watermark w into S. If it is a n-bin watermark, then $w = \text{cod}(m, \boldsymbol{p}_{\text{cod}})$ is computed in advance. The average, maximal and minimal transparency of E is measured ($\text{tra}_{E\text{ave}}$, $\text{tra}_{E\max}$ and $\text{tra}_{E\min}$) by computing the Objective Difference Grade

(ODG) [14] with the implementation of [25]. Furthermore, the detection/retrieval function tries to detect w or to retrieve m' after applying the embedding function in order to measure the detection success det_D or det_R and the retrieval capacity. After a successful embedding, the marked audio signal $\hat{S}$ is attacked by single attacks $A_{i,j}$ and its default attack parameters $\boldsymbol{p}_{A_{i,j}}$ provided by StirMark for Audio (SMBA) [36]. The average, maximal and minimal attacking transparency ($\mathrm{tra}_{A\mathrm{ave}}$, $\mathrm{tra}_{A\mathrm{max}}$ and $\mathrm{tra}_{E\mathrm{min}}$) of $A_{i,j}$ with $\boldsymbol{p}_{A_{i,j}}$ is measured. Then, the watermark detector D tries to detect w and depending on the watermark algorithm, the retrieval function R retrieves m' from $\tilde{S}$. For that, the parameters $\boldsymbol{p}_D$ are used for D and $\boldsymbol{p}_R$ for R. The following Figure 2 shows the test scenario and introduces the simple measuring points.

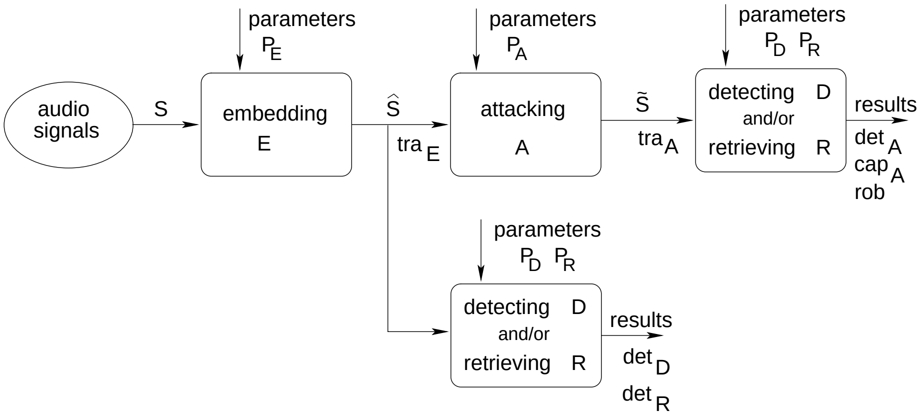

Fig. 2. Test Environment

The detection/retrieval function measures det_D and det_R and its derived average values $\mathrm{det}_{D\mathrm{ave}}$ and $\mathrm{det}_{R\mathrm{ave}}$ after embedding. If this value is ≈ 0.00, which indicates, that the embedding fails for all given audio files, then the attacking and the measurement of its derived values ($\mathrm{rob}_{\mathrm{ave}}^{\mathrm{byte}}$, cap_A, $\mathrm{det}_{A\mathrm{ave}}$ and $\mathrm{det}_{A\tau\mathrm{ave}}$) also does not provide usable test results. A possible reason is, that the cover signal S does not provide enough marking positions for w, which means, that m cannot be embedded completely. If this happens in our tests, then we firstly deduce hat m does not fits into S. Secondly, we obtain the retrieved capacity and based on this value, the attacking capacity and robustness is measured.

Embedding Function:
For the embedding function ($\hat{S} = E(S, cod(m, \boldsymbol{p}_{cod}), \boldsymbol{p}_E)$) the parameters are introduced in the following. For $\boldsymbol{p}_E$, we use default and/or transparency optimized parameters to provide a comparability with respect to robustness and capacity [21, 27]. Thereby, Table 4 shows the used embedding parameters setting. If Ω needs m, then for all tests m=" *Tests*" with $\mathrm{cap}_E^*(\text{“}Tests'') = 40$ bits

$\hat{=}$ 5 bytes. Thereby, for all test, a fixed embedding capacity (cap_E) is used. If a secret key is needed, then mostly $k = 1234$ is used. The following Table 8 summarizes the embedding and evaluation setting.

Table 8. Used embedding parameters for the measure functions.

embedding and evaluation setting	value
m	"Tests"
cap_E^*	40 bits $\hat{=}$ 5 bytes

The value $\tau = 0.7$ ensures, that only attacks with $\mathrm{tra}_{A\mathrm{rel}} > 0.7$ are able to destroy the watermark successful. The other attacks does not achieve the requested quality. The fix set value $\nu = 0.7$ defines, that a retrieved message m' is destroyed, when at least 70% of the retrieved message is false.

Attacking Function:
The attacking function A uses all of the single attacks provided by SMBA. Therefore, 42 different single attacks are used on $\hat{S}$. Firstly, the attacks run with their default parameters ($\boldsymbol{p}_{Ai,2}$) from SMBA and secondly, the parameters for the attacks are changed twice to optimize the attacking strength and attacking transparency ($\boldsymbol{p}_{Ai,1}, \boldsymbol{p}_{Ai,3}$) and thirdly, another set of attacking functions ($\boldsymbol{p}_{Ai,4}$) is used. The following Table 9 shows the attacks and their used parameter settings. The first column shows the name of the single attack. The second, third and fourth column show the attacks and its attacking parameters in use. $\boldsymbol{p}_{Ai,1}$ and $\boldsymbol{p}_{Ai,2}$ contain 29 attacks and $\boldsymbol{p}_{Ai,3}$ contains 26 attacks. The fifth column shows the 13 attacks for $\boldsymbol{p}_{Ai,4}$, which only contain the attacks with unchanged parameters for attack tuning, marked with $\varnothing$. An empty cell means, that this attack does not exist in the attacking set.

For our inter–algorithm evaluation, we use all of the attacks $\boldsymbol{p}_{Ai,(1,\ldots,4)}$ to provide a large attacking set $\mathcal{A}$. In contrast, the attacking set is split into four attacking sets $\boldsymbol{p}_{Ai,1}$, $\boldsymbol{p}_{Ai,2}$, $\boldsymbol{p}_{Ai,3}$ and $\boldsymbol{p}_{Ai,4}$ with $i = \{1, \ldots 42\}$ for our intra–algorithm evaluation and analysis.

The transparency of E and A is measured by computing the ODG value of $\mathrm{tra}_{E\mathrm{rel}}(\Omega^*, S)$ and $\mathrm{tra}_{A\mathrm{rel}}(\Omega^*, \hat{S}, \tilde{S})$, which is needed to identify the transparency success of $A_{i,j}$. The measured ODG values are in the range of $[-4, 0]$ (where -4 is the worst, -3 is bad, -2 is good, -1 is better and 0 the best) and are scaled into a range of $[0, 1]$ by computing: $\left(1 - \frac{[-4,0]}{-4}\right) \to [0, 1]$.

Detection/Retrieval Function:
The detection function $D(\tilde{S}, \boldsymbol{p}_E, \boldsymbol{p}_D, \boldsymbol{p}_{cod}, |S, m|) \to \{0, 1\})$ of the evaluated watermarking algorithms tries to detect w and, if possible, $R(\tilde{S}, \boldsymbol{p}_E, \boldsymbol{p}_R, [S, m])$ tries to retrieve m' from $\tilde{S}$. Therefore, we count the number of positive detected w ($det_{D\mathrm{ave}}$)and correctly retrieved m' ($\mathrm{det}_{R\mathrm{ave}}$) and measure the capacity $\mathrm{cap}_{R\mathrm{rel}}^*$ and det_A and $\mathrm{det}_{A\mathrm{ave}}$. If the retrieved capacity after embedding is lower than the

Table 9. Used attacking parameters $\boldsymbol{p}_{A\,i,j}$, $(i = \{1, \ldots, 42\}, j = \{1, \ldots, 4\})$

attack $A_{i,j}$	$\boldsymbol{p}_{A\,i,1}$	$\boldsymbol{p}_{A\,i,2}$	$\boldsymbol{p}_{A\,i,3}$	$\boldsymbol{p}_{A\,i,4}$
$A_{1,j}$=AddBrumm	2000,55	2500,55	3000,55	
$A_{2,j}$=AddDynNoise	10	20	30	
$A_{3,j}$=AddFFTNoise	1024,20000	1024,30000	1024,40000	
$A_{4,j}$=AddNoise	700	1000	1300	
$A_{5,j}$=AddSinus	80,3000	120,3000	130,3000	
$A_{6,j}$=Amplify	80	50	120	
$A_{7,j}$=BassBoost	150,4	150,6.123	150,8	
$A_{8,j}$=BitChanger	100,99.9	100,99.9	1000,99.9	
$A_{9,j}$=Compressor	6.123,1.5	6.123,2.1	6.123,3.5	
$A_{10,j}$=CopySample	10000,20,6000	10000,2000,6000	10000,200,6000	
$A_{11,j}$=CutSamples	100,7	1000,7	10000,7	
$A_{12,j}$=DynamicPitchScale	0.6,3,32000,64000	1.4,3,32000,64000	1.1,3,32000,64000	
$A_{13,j}$=DynamicTimeStretch	0.6,3,32000,64000	1.4,3,32000,64000	1.1,3,32000,64000	
$A_{14,j}$=Echo	20	2000	200	
$A_{15,j}$=Exchange				∅
$A_{16,j}$=ExtraStereo	10	20	30	
$A_{17,j}$=FFT_HLPassQuick	1024,150,13000	1024,300,15000	1024,150,15000	
$A_{18,j}$=FFT_Invert				1024
$A_{19,j}$=FFT_RealReverse				1024
$A_{20,j}$=FFT_Stat1				1024
$A_{21,j}$=FlippSample	10000,20,6000	10000,2000,6000	10000,200,6000	
$A_{22,j}$=Invert				∅
$A_{23,j}$=LSBZero				∅
$A_{24,j}$=Noise_Max	23,1365,200	23,1365,300	23,1365,400	
$A_{25,j}$=Normalizer1	2048,28000,1	2048,28000,0		
$A_{26,j}$=Normalizer2	2048,28000,1,2500	2048,28000,0,2500		
$A_{27,j}$=Nothing				∅
$A_{28,j}$=Pitchscale	0.95	1.05	1.01	
$A_{29,j}$=RC_HighPass	70	150	300	
$A_{30,j}$=RC_LowPass	12000	15000	17000	
$A_{31,j}$=ReplaceSamples	20,1.5	150,1.5	525,1.5	
$A_{32,j}$=Resampling	11025	22050		
$A_{33,j}$=Smooth				∅
$A_{34,j}$=Smooth2				∅
$A_{35,j}$=Stat1				∅
$A_{36,j}$=Stat2				∅
$A_{37,j}$=TimeStretch	0.95	1.05	1.01	
$A_{38,j}$=VoiceRemove				∅
$A_{39,j}$=ZeroCross	100	1000	3000	
$A_{40,j}$=ZeroLength1	5	10	50	
$A_{41,j}$=ZeroLength2	5	10	50	
$A_{42,j}$=ZeroRemove				∅

embedding capacity for all audio signals, then the given watermarking scheme does not provide enough marking positions for m. In this case, the size of $|m'|$, which is similar to the maximum possible embedding capacity for the given test set, is used for the following measurements.

3.3 Test Goals

The introduced test scenario from Table 3 is used to evaluate and compare the selected watermarking algorithms with inter– and intra–algorithm evaluation and analysis and to show the usage of the predefined theoretical framework. 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 w and/or the retrieveability of m' in $\hat{S}$ and $\tilde{S}$ 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 summary shows the test goals together with the measured parameters.

- Embedding function:
  - Embedding transparency: $\mathrm{tra}_{E\mathrm{ave}}$, $\mathrm{tra}_{E\max}$ and $\mathrm{tra}_{E\min}$
  - Embedding capacity: cap^*_E, a given fixed value
- Detection/Retrieval function:
  - Detection/Retrieval success: $\mathrm{det}_{D\mathrm{ave}}$ only for Ω_4 and $\mathrm{det}_{R\mathrm{ave}}$, $\mathrm{det}_{R\tau\mathrm{ave}}$ and $\mathrm{cap}^*_{R\mathrm{rel}}$ for $\Omega_1, \Omega_2, \Omega_3$ and Ω_5
- Attacking function:
  - Attacking transparency: $\mathrm{tra}_{A\mathrm{ave}}$, $\mathrm{tra}_{A\max}$ and $\mathrm{tra}_{E\min}$
  - Robustness: $\mathrm{rob}^{\mathrm{byte}}_{\mathrm{ave}}$, $\mathrm{det}_{A\mathrm{ave}}$, $\mathrm{det}_{A\tau\mathrm{ave}}$
  - Attacking capacity: $\mathrm{cap}_{A\mathrm{ave}}$, $\mathrm{cap}_{A\max}$ and $\mathrm{cap}_{E\min}$

These properties are measured on the test set S_{SQAM} and in the following section their results are presented.

4 Test Results

In this section we show and discuss the test results. Therefore, we introduce firstly the test results for the embedding, attacking and detection/retrieval function of each watermarking algorithm to compare them each other with inter–algorithm evaluation. Secondly, we show and discuss the test results for an intra–algorithm analysis, where one watermarking algorithm (Ω^*_1) is evaluated with four different attacking parameter sets.

Inter–algorithm evaluation

For all watermarking algorithms, the embedding function E is used 16 times (because of 16 audio files) and if it is able to successfully embed m into all audio files, then the full attacking set $\mathcal{A}$ is used 97 times (because of the different attacks and different attacking parameters $\boldsymbol{p}_A$). Therefore, the detection and/or retrieving function is also called 1552 times. If the embedding fails, then a minor number of detection/retrieval functions is performed.

$\boldsymbol{\Omega_1}$: This watermarking scheme is able to embed m into all audio files successfully ($\mathrm{det}_{R\mathrm{ave}} = 1.00$). Thereby, a fixed embedding capacity $\mathrm{cap}^*_E = 40$ (bits) is used for embedding and the retrieval function returned $\mathrm{cap}^*_{R\mathrm{rel}} = 40$ for all audio files. The test results for the embedding, retrieval and attacking function are shown in the following Table 10.

Table 10. Test results for Ω^*_1

embedding E	retrieval R	attacking A
cap^*_E=40	$\mathrm{det}_{R\mathrm{ave}}$=1.00, $\mathrm{cap}^*_{R\mathrm{rel}}$=40	$\mathrm{cap}_{A\mathrm{ave}}$=0.80, $\mathrm{cap}_{A\mathrm{min}}$=0.00, $\mathrm{cap}_{A\mathrm{max}}$=1.00, $\mathrm{det}_{A\mathrm{ave}}$=0.44
$\mathrm{tra}_{E\mathrm{ave}}$=0.63, $\mathrm{tra}_{E\mathrm{min}}$=0.02, $\mathrm{tra}_{E\mathrm{max}}$=0.95		$\mathrm{tra}_{A\mathrm{ave}}$=0.38, $\mathrm{tra}_{A\mathrm{min}}$=0.02, $\mathrm{tra}_{A\mathrm{max}}$=1.00
		$\mathrm{rob}^{\mathrm{byte}}_{\mathrm{ave}}$=0.37

The best embedding transparency is measured with $\mathrm{tra}_{E\mathrm{max}} = 0.95$ for the audio test file *spmg54_1.wav* (which is speech) and the worst with $\mathrm{tra}_{E\mathrm{min}} = 0.02$ for audio test file *frer07_1.wav* (which is a single instrument). The average embedding transparency is measured with $\mathrm{tra}_{E\mathrm{ave}} = 0.63$. These measured results shows, that the embedding transparency of Ω^*_1 depends on S and therefore, the quality of E depends on the type of audio content. The retrieval after embedding measured with $\mathrm{det}_{R\mathrm{ave}} = 1.00$ shows, that the given message m fits into all audio files. The test results for the attacking function with the following retrieval show, that Ω^*_1 is not robust against the attacks $A_{18,4}$ and $A_{22,4)}$ (FFT_Invert and Invert), which provides also a attacking transparency $\mathrm{tra}_{A\mathrm{ave}} = 0.72$. If the detection success after attacking function is measured, then Ω^*_1 has a value of $\mathrm{det}_{A\mathrm{ave}} = 0.44$, which means that this watermarking scheme is robust against all performed attacks in about 44% independent of the attacking transparency. In contrast, the average robustness, has a value $\mathrm{rob}^{\mathrm{byte}}_{\mathrm{ave}} = 0.38$.

$\boldsymbol{\Omega_2}$: This watermarking scheme could not embed m into all audio files successfully. A successful detection of m' after embedding failed. Hence, the retrieved capacity after embedding $\mathrm{cap}^*_{R\mathrm{rel}} = 8$ shows that m did not fit into S. Only first few bits are retrievable which implies a low embedding capacity. Therefore, only the first 8 bits of m (“T”) are used by setting $\mathrm{cap}^*_E = 8$ for the following measures of robustness and attacking capacity. Furthermore, Ω^*_2 could only embed the watermark into 12 audio files. For *frer07_1.wav*, *gspi35_2.wav*, *gspi35_2.wav* and *horn23_2.wav* (which are all single instruments) the embedding of any bits fails and these files are excluded from the test set. In this case, where $m' \neq m$ is not retrievable for any S of S_{SQAM}, $\text{det}_{R\text{ave}} = 0.00$. This shows, that the given watermarking scheme does not provide enough marking positions. In our exemplary test evaluation, the measured retrieval capacity $\text{cap}^*_{R\text{rel}} = 8$ bits is measured. Therefore, the following measures are based on the lower retrieved capacity of 8 bits needed for the normalization. The following Table 11 shows the test results for the Ω^*_2 watermarking scheme.

Table 11. Test results for Ω^*_2

embedding E	retrieval R	attacking A
cap^*_E=40	$\text{det}_{R\text{ave}}$=0.00, $\text{cap}^*_{R\text{rel}}$=8	$\text{cap}_{A\text{ave}}$=0.19, $\text{cap}_{A\text{min}}$=0.00, $\text{cap}_{A\text{max}}$=0.20, $\text{det}_{A\text{ave}}$=0.00
$\text{tra}_{E\text{ave}}$=0.66, $\text{tra}_{E\text{min}}$=0.10, $\text{tra}_{E\text{max}}$=0.95		$\text{tra}_{A\text{ave}}$=0.43, $\text{tra}_{A\text{min}}$=0.02, $\text{tra}_{A\text{max}}$=1.00
		$\text{rob}^{\text{byte}}_{\text{ave}}$=0.91

The best embedding transparency is achieved with $\text{tra}_{E\text{max}} = 0.947$ for the test file *frer07_1.wav* (which is a single instrument) and the worst with $\text{tra}_{E\text{min}} = 0.10$ for test file *gspi35_1.wav* (which is a single instrument). The average embedding transparency is 0.66 over S_{SQAM}. The detection success is measured with $\text{det}_{R\text{ave}} = 0.00$ because of the impossibility to embed of the complete lengths of $|m|$. Therefore, the average attacking capacity is $\text{cap}_{A\text{ave}} = 0.19$ and shows, that this watermarking scheme has a low embedding capacity for the given test set and given embedding parameters. The results for the robustness measure, normalized on the correctly embedded 8 bits shows, that the robustness measured on bit and byte errors of Ω^*_2 is high $\left(\text{rob}^{\text{byte}}_{\text{ave}} = 0.91\right)$ It means, that only the attacks with a worse transparency results are able to destroy the watermark successful. Selected attacks are $A_{12,(1,2,3)}$, $A_{13,(1,2,3)}$, $A_{28,(1,2,3)}$, $A_{37,(1,2,3)}$ and $A_{10,(1,2,3)}$, $A_{41,(1,2,3)}$ and $A_{41,(1,2,3)}$ which have a worse attacking transparency.

$\mathbf{\Omega_3}$: This watermarking scheme was not able to embed m into all audio files successfully. The audio file *frer07_1.wav* (which is single instrument) did not provide marking positions for w in S and a retrieval of $m' = m$ directly after the embedding was not successful. Therefore, the average retrieval success is $\text{det}_{R\text{ave}} = \frac{15}{16} = 0.94$. In the following evaluations, this audio file is neglected.

The following Table 12 shows the test results for the Ω_3^* watermarking scheme, excluding this audio file.

Table 12. Test results for Ω_3^*

embedding E	retrieval R	attacking A
cap_E^*=40	$\text{det}_{R\text{ave}}$=0.94, $\text{cap}_{R\text{rel}}^*$=40	$\text{cap}_{A\text{ave}}$=0.35, $\text{cap}_{A\text{min}}$=0.00, $\text{cap}_{A\text{max}}$=1.00, $\text{det}_{A\text{ave}}$=0.31
$\text{tra}_{E\text{ave}}$=0.11, $\text{tra}_{E\text{min}}$=0.02, $\text{tra}_{E\text{max}}$=0.37		$\text{tra}_{A\text{ave}}$=0.41, $\text{tra}_{A\text{min}}$=0.02, $\text{tra}_{A\text{max}}$=1.00
		$\text{rob}_{\text{ave}}^{\text{byte}}$=0.11

The best embedding transparency is measured with $\text{tra}_{E\text{max}} = 0.37$ for the test file *harp40_1.wav* (which is a single instrument) and the worst with $\text{tra}_{E\text{min}} = 0.02$ for test file *gspi35_1.wav* (which is a single instrument). The average embedding transparency is measured with $\text{tra}_{E\text{ave}} = 0.11$, which is worse. These results show, that the used embedding parameters $\boldsymbol{p}_E$ could be tuned and/or the used audio set S_{SQAM} changed to provide better test results for tra_E. This watermarking scheme embeds m multiple times into S. Therefore, m was embedded successfully 1 times ($p_{\max} = 1$) in the following 7 audio files: *gspi35_2.wav*, *harp40_1.wav*, *horn23_2.wav*, *trpt21_2.wav*, *bass47_1.wav*, *quar48_1.wav* and *sopr44_1.wav*[5]. In contrast, m was two times successfully embedable $p_{\max} = 2$ for the following 7 audio files: *gspi35_1.wav*, *spfe49_1.wav*, *spff51_1.wav*, *spfg53_1.wav*, *spme50_1.wav*, *spmf52_1.wav* and *spmg54_1.wav*. For only one audio file (*vioo10_2.wav*) m was successfully embedded three times ($p_{\max} = 3$). The robustness of Ω_3^* is measured with a value of $\text{rob}_{\text{ave}}^{\text{byte}} = 0.11$, whereby the attacks $A_{35,4}$ and $A_{7,(1,2,3)}$ are successful and with good attacking transparency for an attacker.

$\boldsymbol{\Omega_4}$: This watermarking scheme also failed to embed m into all audio files successfully. For four audio files (*frer07_1.wav*, *gspi35_1.wav*, *gspi35_2.wav* and *vioo10_2.wav*) it was not possible to compute the correlation value of the embedded w successful. Therefore, the average detection successful rate is $\text{det}_{D\text{ave}} = \frac{12}{16} = 0.75$ and these audio files are excluded for the following measures. The following Table 13 shows the test results for the Ω_4^* watermarking scheme. If the correlation is reobtained by 70% ($\tau > 0.7$) or more, then the watermark is positive detectable.

[5] For its classification of the audio content, please see Table 7.

Table 13. Test results for Ω_4^*

embedding E	retrieval R	attacking A
	$\mathrm{det}_{D\,\mathrm{ave}}$=0.75, $\mathrm{cap}^*_{R\,\mathrm{rel}}$=$\varnothing$	$\mathrm{det}_{A\,\mathrm{ave}}$=0.70
$\mathrm{tra}_{E\,\mathrm{ave}}$=0.29, $\mathrm{tra}_{E\,\mathrm{min}}$=0.02, $\mathrm{tra}_{E\,\mathrm{max}}$=0.85		$\mathrm{tra}_{A\,\mathrm{ave}}$=0.45, $\mathrm{tra}_{A\,\mathrm{min}}$=0.02, $\mathrm{tra}_{A\,\mathrm{max}}$=1.00
		$\mathrm{rob}^{\mathrm{byte}}_{\mathrm{ave}}$=0.60

The best embedding transparency is measured with $\mathrm{tra}_{E\,\mathrm{max}} = 0.85$ for the test file *spmg54_1.wav* (which is speech) and the worst with $\mathrm{tra}_{E\,\mathrm{min}} = 0.02$ for test file *gspi35_1.wav* (which is a single instrument). The average embedding transparency with $\mathrm{tra}_{E\,\mathrm{ave}} = 0.45$ is measured over S_{SQAM}. The robustness is measured with $\mathrm{rob}^{\mathrm{byte}}_{\mathrm{ave}} = 0.60$, whereby the watermark can be destroyed with the attacks $A_{6,i}$ (Amplify), $A_{22,4}$ (Invert) and $A_{18,4}$ (FFT_Invert), which have an average attacking transparency only of $\mathrm{tra}_{A\,\mathrm{ave}} = 0.84$. The test results for the attacking capacity is not provided by this watermarking algorithm, because no message was embedded (zero-bit watermarking scheme).

$\boldsymbol{\Omega_5}$: This watermarking scheme was also not able to embed m into all audio files successfully. Only for the audio files *gspi35_1.wav* and *gspi35_2.wav* it was not possible to retrieve m' directly after embedding. Therefore, the average retrieval after embedding is measured with $\mathrm{det}_{R\,\mathrm{ave}} = \frac{14}{16} = 0.87$. This watermarking scheme embeds m multiple times, but it was not able to measure the number of multiple embedding p_{max}. The following Table 14 shows the test results for the Ω_5^* watermarking scheme.

Table 14. Test results for Ω_5^*

embedding E	retrieval R	attacking A
cap^*_E=40	$\mathrm{det}_{R\,\mathrm{ave}}$ =0.87, $\mathrm{cap}^*_{R\,\mathrm{rel}}$=40	$\mathrm{cap}_{A\,\mathrm{ave}}$=0.70, $\mathrm{cap}_{A\,\mathrm{min}}$=0.29, $\mathrm{cap}_{A\,\mathrm{max}}$=1.00, $\mathrm{det}_{A\,\mathrm{ave}}$=0.28
$\mathrm{tra}_{E\,\mathrm{ave}}$=0.49, $\mathrm{tra}_{E\,\mathrm{min}}$=0.09, $\mathrm{tra}_{E\,\mathrm{max}}$=0.73		$\mathrm{tra}_{A\,\mathrm{ave}}$=0.36, $\mathrm{tra}_{A\,\mathrm{min}}$=0.02, $\mathrm{tra}_{A\,\mathrm{max}}$=1.00
		$\mathrm{rob}^{\mathrm{byte}}_{\mathrm{ave}}$=0.08

The best embedding transparency is measured with $\mathrm{tra}_{E\,\mathrm{max}} = 0.73$ for the test file *sopr44_1.wav* (which is speech) and the worst with $\mathrm{tra}_{E\,\mathrm{min}} = 0.09$ for test file *frer_1.wav* (which is a single instrument). The average embedding transparency for Ω_5^* is $\text{tra}_{E\text{ave}} = 0.49$. The robustness is measured with 0.08 and the average attacking capacity with 0.70.

Summarizing of the inter–algorithm evaluation and analysis

The test results for the inter–algorithm evaluation and analysis of the five selected watermarking schemes can be summarized as follows. The average embedding transparency $\text{tra}_{E\text{ave}}$ is one of the main properties for the evaluation of watermarking algorithms. Its quality depending on the embedding function has a major importance regarding the watermark application field. As the results show, the measured average embedding transparency differs due to the watermarking schemes. Depending on the embedding parameters and/or the used audio test set, the embedding transparency can be tuned for a specific application field. In contrast, the results of det_D and det_R show that either a given audio signal provides enough marking positions to embed the watermark or not. Therefore, a watermarking scheme, which has a low embedding capacity for an application field, can be identified or it can be seen that not all audio files can be marked. Furthermore, the inter–algorithm analysis results show, that the embedded watermark can be destroyed easily with a specific $A_{i,j}$, but mostly, the attacking transparency is worse. A successful removal or disabling of the embedded watermark without audible distortion less than 0.7 is difficult. For Ω_1^* only the two attacks $A_{18,4}$ and $A_{22,4}$ achieve this requirement. Ω_2^* was robust against the attacks with a good attacking transparency whereby the robustness measure increases, but can only measured for the embedded 8 bits. Ω_3^* and Ω_5^* have similar robustness results in our test environment. For Ω_4^* an attacker could destroy the watermark with the attacks $A_{6,1}$, $A_{18,4}$ or $A_{22,4}$ without audible distortions. Our attacking test set shows, that some attacks with $\text{tra}_{A\text{rel}} < 0.7$ have the power to destroy the watermark successful. Therefore, the attacking set or its parameters need to be tuned. The attacking capacity yielded test results, where the watermark can be destroyed without focus on the attacking transparency and possible embedding capacity. This measured value is only useful for n-bit watermarking schemes, because for zero-bit watermarking no embedding and no attacking capacity is available. In our tests we show, that Ω_1^* has the best and Ω_2^* the worse test results for $\text{cap}_{A\text{ave}}$ as the retired $\text{cap}^*_{R\text{rel}}$ which is already only 8bits as mention on page 4.

In the following Table 15 our three selected properties (embedding transparency, attacking capacity and robustness) of the evaluated watermarking algorithms are shown and exemplary discussed to summarize their performance in the triangle of transparency, capacity and robustness.

It is shown that the inter–algorithm evaluation analysis comparing all five watermarking schemes provides different test results, and the evaluated properties of them are different when measured on the same audio test and the same attacks with its attacking parameter setting $\boldsymbol{p}_{Ai,(1,\ldots,4)}$. The embedding transparency differs from 0.11 for Ω_3^* up to 0.66 for Ω_2^*. For the average attacking

Table 15. Summarized test results with a fixed capacity of given cap_E^*=40 bits

watermarking scheme	$\mathrm{tra}_{E\mathrm{ave}}$	$\mathrm{cap}_{A\mathrm{ave}}$	$\mathrm{rob}_{\mathrm{ave}}^{\mathrm{bit}}$
Ω_1^*	0.63	0.80	0.37
Ω_2^*	0.66	0.19	0.91
Ω_3^*	0.11	0.31	0.11
Ω_4^*	0.29	∅	0.60
Ω_5^*	0.49	0.70	0.08

capacity $\mathrm{cap}_{A\mathrm{ave}}$ the test results show, that Ω_2^* has the lowest (0.19) and Ω_1^* the highest (0.80) retrieved capacity after attacking. For Ω_4^* this value is zero, due to its characteristic as a zero-bit watermarking scheme. The highest robustness is provided by Ω_2^* 0.91, which means, that the watermark can only be destroyed with audible distortions. It is assumed, that the price for the high robustness is the low embedding capacity. The attacking transparency is very important for successful attacking. For Ω_1^* the measure robustness is 0.37. The both algorithms Ω_3^* and Ω_5^* have a similar robustness, which is 0.11 and 0.08.

The test results $\mathrm{tra}_{E\mathrm{ave}}$, $\mathrm{cap}_{A\mathrm{ave}}$ and $\mathrm{rob}_{\mathrm{ave}}$ are exemplary used to discuss and visualize the position of $\Omega_{1,\ldots,5}$ in the triangle. In this visualization, the position inside the triangle depends on the ratio between the values of the corners and it is not simple to identify the exact position. If for example a watermarking scheme is bad for all thee properties, then it has the same position as a watermarking scheme, which is good for all three properties. Furthermore, if the position of a watermarking algorithm is directly located in one corner of the triangle, then this value must be 1.00 and the other two values must be 0.00. Another example to introduce the problems with the triangle is, that for e.g. a watermarking scheme, for which is measured 0.4 for all three properties has the same position as another watermarking scheme with all three values measure of 0.7. Our solution is to identify the best position depending on the ratio between the tree properties. Therefore, the exact values of transparency, capacity and robustness are charted by drawing the values on the bisecting line of an angle, whereby the value 1.00 is the corner. After charting the three properties of an algorithm, the centroid of the resulting triangle is the relative position of the watermarking scheme. Our idea a is to introduce the effect of the measured values for $\mathrm{tra}_{E\mathrm{ave}}$, $\mathrm{rob}_{\mathrm{ave}}^{\mathrm{bit}}$ and $\mathrm{cap}_{A\mathrm{ave}}$, whereby points of the exact measured position are shown for each $\Omega_{1,\ldots,5}$. We selected the symbol ■ for Ω_1^*, the symbol + for Ω_2^*, the symbol ▲ for Ω_3^*, the symbol × for Ω_4^* and the symbol ⧫ for Ω_5^*. These symbols are drawn on the bisecting lines of the angles with the corner having the value of 1.00 and the opposite 0.00. The positions of $\Omega_{1,\ldots,5}$ are marked with the selected symbol and an enclosed circle ●.

Figure 3 shows the approximate "position" of the compared watermarking schemes.

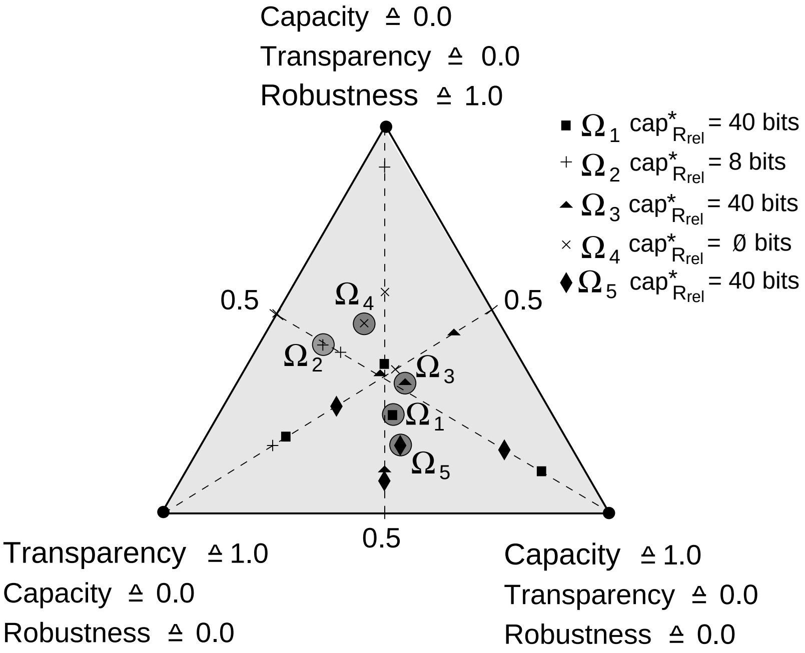

Fig. 3. Test results with attacks in the triangle with a fixed embedding capacity (inter–algorithms evaluation) with the selected measure $\mathrm{tra}_{E\mathrm{ave}}$, $\mathrm{cap}_{A\mathrm{ave}}$ and $\mathrm{rob}^{\mathrm{bit}}_{\mathrm{ave}}$

The overall goal was to summarize the results exemplarily and from the figure it should be clear, that only the three selected measures can be visualized and not all results can be shown like the number of audio files where no watermark could be embedded.

Intra–algorithm evaluation

In the following, we introduce test results for an intra–algorithm evaluation and analysis of Ω^*_1 where four different attack sets and its attacking parameters $\boldsymbol{p}_{Ai,(1,\ldots,4)}$ are used. The test results only for Ω^*_1 with these four attacking sets are exemplary summarized in the following Table 16 and introduced as follows: The test results for the embedding transparency and attacking capacity are inherited from the inter–algorithm evaluation. The robustness is measured with the four attacking sets $\boldsymbol{p}_{Ai,1}$, $\boldsymbol{p}_{Ai,2}$, $\boldsymbol{p}_{Ai,3}$ and $\boldsymbol{p}_{Ai,4}$. Thereby it is measured, that the robustness is $\mathrm{rob}^{\mathrm{bit}}_{\mathrm{ave}} = 1.00$ for the three attacking sets $\boldsymbol{p}_{Ai,1}$, $\boldsymbol{p}_{Ai,2}$ and $\boldsymbol{p}_{Ai,3}$. Thereby, it is identified, that the attacks in these attacking sets, which destroy the watermark successful do not have $\mathrm{tra}_{A\mathrm{rel}} > 0.7$ which is required for a successful attack. The fourth attacking set $\boldsymbol{p}_{Ai,4}$ includes the attacks $A_{18,4}$ (FFT_Invert) and $A_{22,4}$ (Invert) which destroyed the watermark successfully ($\nu > 0.7$ and $\mathrm{tra}_{A\mathrm{rel}} > 0.7$). Thereby, $\boldsymbol{p}_{Ai,1}$, $\boldsymbol{p}_{Ai,2}$ and $\boldsymbol{p}_{Ai,3}$ have the same measured properties and are located at the same position in the triangle. The robustness of the attacking set $\boldsymbol{p}_{Ai,4}$ differs from the three previous attacking sets and therefore also its position differs in the triangle.

Table 16. Summarized test results for the intra algorithm evaluation of Ω_1^*

watermarking scheme	$\text{tra}_{E\text{ave}}$	$\text{cap}_{A\text{ave}}$	$\text{rob}_{\text{ave}}^{\text{bit}}$
$\boldsymbol{p}_{Ai,1}$	0.63	0.80	0.87
$\boldsymbol{p}_{Ai,2}$	0.63	0.80	0.93
$\boldsymbol{p}_{Ai,3}$	0.63	0.80	0.99
$\boldsymbol{p}_{Ai,4}$	0.63	0.80	0.06

To visualize the intra–algorithm evaluation and analysis result for Ω_1^* with four different attacking parameter sets, the following Figure 4 shows the "position" in the triangle of transparency, capacity and robustness. Thereby the same idea (introduced for the inter–algorithm evaluation) to visualize the triangle is used. It is shown, that $\boldsymbol{p}_{Ai,1}$ (■), $\boldsymbol{p}_{Ai,2}$ (+) and $\boldsymbol{p}_{Ai,3}$ (▲) have the same position in the triangle and the fourth $\boldsymbol{p}_{Ai,4}$ (×) differs only a little with regarding to robustness.

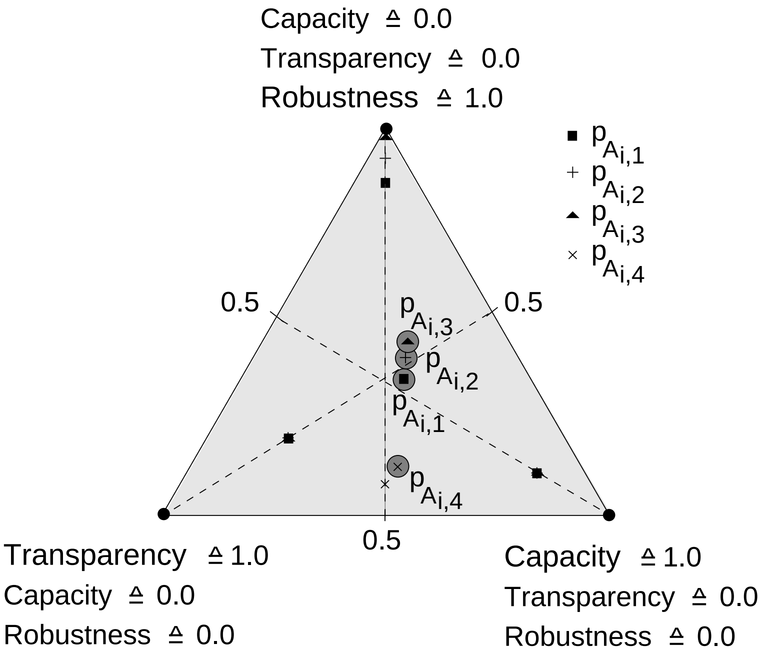

Fig. 4. Test results after attacking for Ω_1^* in the triangle with four different $\boldsymbol{p}_{Ai,(1,\ldots,4)}$ (intra–algorithms evaluation) with the selected measure $\text{tra}_{E\text{ave}}$, $\text{cap}_{A\text{ave}}$ and $\text{rob}_{\text{ave}}^{\text{bit}}$

Therefore it could be shown, that a different attacking sets of $\boldsymbol{p}_{Ai,(1,\ldots,4)}$ effects the measured main properties of a given watermarking scheme. Depending on these sets the properties of the watermarking scheme differ and therefore the appropriate application field might change for the usage of the algorithm. Therefore, the best and worst test results of watermarking schemes depend on the parameter settings used for embedding, attacking and detection/retrieval and the test set used for evaluation.

5 Conclusion and Future Work

In this paper, in the first part we have presented a theoretical framework to provide a description and formalization of the application oriented properties robustness, transparency and capacity of digital watermarking algorithms. Therefore, the embedding, detecting/retrieval and attacking functions with its parameters and the main properties robustness, transparency and capacity are defined. Furthermore, the dependencies between these properties are discussed. To provide evaluation and comparison of watermarking algorithms, measuring methods are derived. Our introduced methodology is easily enhanced by defining derived or new evaluation measures. The idea is to normalize the measures into the triangle within the resolution between 0 and 1.

The second part of the paper has presented a practical usage of the predefined theoretical framework by using selected audio watermarking algorithms for comparing it in the triangle of the main properties with a selected set of defined measures. The evaluation of the algorithms shows that the embedding and detection/ retrieval parameters are different and provide different test results. To allow a comparison we used a fixed test set of 16 audio files and a fixed embedding capacity. The test results for the five watermarking algorithms show that the inter–algorithm evaluation based on a given fixed set of audio files and a given fixed set of attacking parameters provides different results for the watermarking schemes. On the one hand, one watermarking algorithm — a zero-bit watermarking scheme — provides a high robustness, but the transparency of the embedding function is bad. On the other hand, the *n*-bit watermarking schemes provides good transparency results for the embedding function, but the robustness decreases.

The test results for an intra–algorithm analysis show, that a different attacking parameter setting changes the measured properties of a watermarking scheme. Therefore, the potential attack scenario of a used application field for a watermarking algorithm should be known or estimated before applying the scheme.

We hope that the introduced methodology will be widely used to allow a more precise comparison of watermarking algorithms and their test results. The method has room for enhancement and should be seen as attempt to specify a normalized measure for a more precise inter–algorithm and intra–algorithm evaluation and analysis. Future work is to enhance our theoretical framework with other properties of watermarking algorithms (like fragility for integrity evaluation, security or complexity) and to compare watermarking schemes with different embedding parameter settings. Furthermore, the 2-dimensional geometric triangle should be replaced by another geometric figure (3-dimensional) which provides more and detailed space for the position of a watermarking scheme.

5.1 Acknowledgement

The work about the watermarking algorithms and single attacks 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.

Effort for the Audio WET system, which was used for the evaluation, is sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number FA8655-04-1-3010. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

UOC authors are partially supported by the Spanish MCYT and the FEDER funds under grants MULTIMARK TIC2003-08604-C04-04 and PROPRIETAS-WIRELESS SEG2004-04352-C04-04

With respect to the experimental evaluations, this publication has been produced partly with the assistance of the EU-India cross cultural program (project CultureTech, see [7]). The content of this publication is the sole responsibility of the University Magdeburg and their co-authors and can in no way be taken to reflect the views of the European Union.

Furthermore, we thank Jörg Wissen and Raquel Buil who helped us to set up the test environment and to analyze the test results.

References

1. *Evaluation of Natural Language Processing Systems*, FINAL REPORT, EAGLES DOCUMENT EAG-EWG-PR.2, Version of September 1995, section Methods for System Measurement, http://www.issco.unige.ch/ewg95/, May 2006
2. B. Macq, J. Dittmann and E. J. Delp, *Benchmarking of Image Watermarking Algorithms for Digital Rights Management*, Proceedings of the IEEE, Special Issue on: Enabling Security Technology for Digital Rights Management, pp. 971–984, Vol. 92 No. 6, June 2004
3. F. Cayre, C. Fontaine and T. Furon, *Watermarking security, part I: theory*, In: Security, Steganography and Watermarking of Multimedia Contents VII, Ping Wah Wong, Edward J. Delp III, Editors, Proceedings of SPIE Vol. 5681, San Jose, USA, 2005

4. F. Cayre, C. Fontaine and T. Furon, *Watermarking security, part II: practice*, In: Security, Steganography and Watermarking of Multimedia Contents VII, Ping Wah Wong, Edward J. Delp III, Editors, Proceedings of SPIE Vol. 5681, San Jose, USA, 2005
5. Checkmark Benchmarking, *http://watermarking.unige.ch/Checkmark/*, 2006
6. I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, *Secure spread spectrum watermarking for multimedia*, IEEE Transactions on Image Processing, 6(12):1673–1687, 1997.
7. The Culture Tech Project, Cultural Dimensions in digital Multimedia Security Technology, a project funded under the EU-India Economic Cross Cultural Program, *http://amsl-smb.cs.uni-magdeburg.de/culturetech/*, requested July 2005
8. J. Dittmann, *Digitale Wasserzeichen*, Berlin, Springer, Xpert.press, 2000, ISBN 3-540-66661-3
9. J. Dittmann, M. Steinebach, A. Lang and S. Zmudizinski, *Advanced audio watermarking benchmarking*, Security, Steganography, and Watermarking of Multimedia Contents VI, Edward J. Delp III, Ping W. Wong (Eds.) SPIE Vol. 5306, SPIE and IS&T, pp. 224-235, Electronic Imaging Science and Technology, 19-22 Jan. 2004, San Jose, California, USA, ISBN 0-8194-5209-2, 2004
10. J. Domingo-Ferrer and J. Herrera-Joancomartí. Simple collusion-secure fingerprinting schemes for images. In *Proceedings of the Information Technology: Coding and Computing ITCC'2000*, pages 128–132. IEEE Computer Society, 2000.
11. R. Dugad, K. Ratakonda and N. Ahuja, *A New Wavelet-Based Scheme for Watermarking Images*, IEEE International Conference on Image Processing, Chicago, 1998
12. J. Fridrich, *Applications of data hiding in digital images*, Tutorial for the ISPACS 1998 conference in Melburne, Australia, 1998
13. H. Inoue, A. Miyazaki, A. Yamamoto and T. Katsura, *A Digital Watermarking Technique Based on the Wavelet Transform and Its Robustness on Image Compression and Transformation*, IEICE Trans. Fundamentals, vol. E82-A, no. 1, 1999
14. ITU-R Recommendation BS.1387, *Method for Objective Measurements of Perceived Audio Quality*, *http://www.itu.int/rec/R-REC-bs/en*, Dec. 1998
15. A.J. Jerri, *The Shannon sampling theorem – its various extensions and application: a tutorial review*, Proc. IEEE, 65, 1565-1597, 1977
16. T. Kalker, *Considerations on watermarking security*, In: Proceedings of the IEEE Multimedia Signal Processing MMSP01 workshop, Cannes, France, pp. 201–206, 2001
17. D. Kirovski and H.S. Malvar, *Spread Spectrum Watermarking of Audio Signals*, IEEE Transactions on Signal Processing, Vol.51, (no.4), pp.1020-33, 2003
18. C. Kraetzer, J. Dittmann and A. Lang, *Transparency benchmarking on audio watermarks and steganography*, to appear in SPIE conference, at the Security, Steganography, and Watermarking of Multimedia Contents VIII, IS&T/SPIE Symposium on Electronic Imaging, 15-19th January, 2006, San Jose, USA, 2006
19. M. Kutter, S. Voloshynovskiy and A. Herrigel, *Watermark copy attack*, In Ping Wah Wong and Edward J. Delp eds., IS&T/SPIEs 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II, Vol. 3971 of SPIE Proceedings, San Jose, California USA, 23-28 January 2000
20. A. Lang and J. Dittmann, *StirMark and profiles: from high end up to preview scenarios*, Virtual Goods 2004, 27-29.05.2004, to appear in http://virtualgoods.tu-ilmenau.de/2004/, Ilmenau, 2004

21. A. Lang and J. Dittmann, *Profiles for Evaluation - the Usage of Audio WET*, to appear in SPIE conference, at the Security, Steganography, and Watermarking of Multimedia Contents VIII, IS&T/SPIE Symposium on Electronic Imaging, 15-19th January, 2006, San Jose, USA, 2006
22. A. Lang and J. Dittmann, *Digital Watermarking of Biometric Speech References: Impact to the EER System Performance*, to appear in SPIE conference, at the Security, Steganography, and Watermarking of Multimedia Contents IX, IS&T/SPIE Symposium on Electronic Imaging, 28th January - 01th February, 2006, San Jose, USA, 2007
23. A. Lang, J. Dittmann, E. T. Lin and E. J. Delp, *Application oriented audio watermark benchmark service*, In: Delp, Edward J. (Hrsg.), Wong, Ping W. (Hrsg.), Security, steganography, and watermarking of multimedia contents VII (Electronic imaging science and technology San Jose, California, USA, 17-20 January 2005), Bellingham, Wash., pp. 275–286, ISBN 0-8194-5654-3, 2005
24. A. Lang, J. Dittmann, R. Spring and C. Vielhauer, *Audio watermark attacks: from single to profile attacks*, In: City University of New York (Veranst.): Multimedia and Security, MM & Sec'05 (Workshop New York, NY, USA August 1-2 2005), New York, NY, ACM, pp. 39–50, ISBN 1-59593-032-9, 2005
25. A. Lerch, zplane.development, *EAQUAL - Evaluation of Audio Quality*, Version: 0.1.3alpha, http://www.mp3-tech.org/programmer/misc.html, 2002
26. libSNDfile library, *http://www.mega-nerd.com/libsndfile/*, May, 2006
27. D. Megías, J. Herrera-Joancomartí, and J. Minguillón, *A robust audio watermarking scheme based on MPEG 1 layer 3 compression*, In Communications and Multimedia Security - CMS 2003, Lecture Notes in Computer Science 2828, pages 226–238, Turin (Italy), October 2003. Springer-Verlag.
28. D. Megías, J. Herrera-Joancomartí, and J. Minguillón. *An audio watermarking scheme robust against stereo attacks*, In Proceedings of the Multimedia and Security Workshop, pages 206–213, Magdeburg (Germany), September 2004. ACM.
29. D. Megías, J. Herrera-Joancomartí, and J. Minguillón. *Robust frequency domain audio watermarking: a tuning analysis*, In International Workshop on Digital Watermarking — IWDW 2004, Lecture Notes in Computer Science 3304, pages 244–258, Seoul (Korea), November 2004. Springer-Verlag.
30. Optimark, *http://poseidon.csd.auth.gr/optimark/*, 2006
31. Luis Pérez-Freire, Pedro Comesaña and Fernando Pérez-González, *Information-Theoretic Analysis of Security in Side-Informed Data Hiding*, Information Hiding, pp. 131–145, 2005
32. F. A. P. Petitcolas, M. Steinebach, F. Raynal, J. Dittmann, C. Fontaine and N. Fates, *Public automated web-based evaluation service for watermarking schemes: StirMark Benchmark*, In: Security and Watermarking of Multimedia Contents III, Ping Wah Wong, Edward J. Delp III, Editors, Proceedings of SPIE Vol. 4314, Bellingham WA, USA, pp. 575–584, ISBN 0-8194-3992-4, 2001.
33. J.M. Shapiro, *Embedded image coding using zerotrees of wavelet coefficients*, IEEE Trans. Signal Processing, vol. 41, no.12, pp. 3445–3462, 1993
34. SQAM — Sound Quality Assessment Material, *http://sound.media.mit.edu/mpeg4/audio/sqam/*, 2006
35. Stirmark Benchmark, *http://www.petitcolas.net/fabien/watermarking/stirmark/*, 2006
36. StirMark Benchmark for Audio, *http://amsl-smb.cs.uni-magdeburg.de/*, 2005
37. C. Vielhauer, T. Scheidat, A. Lang, M. Schott, J. Dittmann, T.K. Basu and P.K. Dutta; *Multimodal Speaker Authentication – Evaluation of Recognition Performance of Watermarked References*; In: Proceedings of MMUA 2006, Toulouse, France, 2006

38. S. Voloshynovskiy et al., *Attacks on Digital Watermarks: Classification, Estimation-Based Attacks, and Benchmarks*, IEEE Communications Magazine, vol. 39(8), pp. 118–126, Aug. 2001
39. Watermark Evaluation Testbed for Audio, *http://audio-wet.cs.uni-magdeburg.de/wet/*, 2006