Application Oriented Audio Watermark Benchmark Service

Andreas Lang[1], Jana Dittmann[1], Eugene T. Lin[2], Edward J. Delp[2]

[1]Otto-von-Guericke University of Magdeburg, Germany
Working group: Advanced Multimedia and Security

{andreas.lang, jana.dittmann}@iti.cs.uni-magdeburg.de

[2]Video and Image Processing Laboratory
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana USA

{linet, ace}@ecn.purdue.edu

Abstract

Methodologies and tools for watermark evaluation and benchmarking facilitate the development of improved watermarking techniques. In this paper, we want to introduce and discuss the integration of audio watermark evaluation methods into the well-known web service Watermark Evaluation Testbed (WET) [1]. WET is enhanced by using. A special set of audio files with characterized content and a collection of single attacks as well as attack profiles [3] will help to select special audio files and attacks with their attack parameters.

Motivation

Digital watermarking has been proposed for a variety of applications, including content protection, authentication, and digital rights management. Many watermarking techniques have made claims regarding performance, such as invisibility, robustness, and capacity but currently it is not easy to objectively evaluate their performance claims. Watermark evaluation methodologies and tools 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.

It is clear that the development of digital watermarks is connected with their evaluation. The performance claimed by watermarking algorithms or their implementations should be verified using a fair and objective process. This process can be very complex and therefore the idea is to describe the evaluation process with special attacks and their specified or selected attack parameters or with a set of signal processing functions, which could be introduced to the content (for example, lossy compression). To improve the evaluation of digital watermarking algorithms the Watermark Evaluation Testbed (WET) system was developed at Purdue University [1][4] featuring a web-based user interface for the evaluation of still-image watermarks. In this paper, we will discuss and describe the procedure to integrate audio evaluation or benchmarking function into the existing WET service and the definition of complex attack scenarios, which are called profile attacks.

The paper is structured as followed: Section 1 is a review of the existing WET benchmark for image watermarking. Section 2 is a review of the StirMark for audio and a description of audio benchmark profiles. The major portion of this section is a detailed description of the profiles, their parameters, and organization. Section 3 describes a proposed method for integrating the profiles into WET for Audio using an XML framework. Lastly, Section 4 has the conclusions and directions for future work.

1 The WET evaluation service

In this section we introduce the Watermark Evaluation Testbed (WET) [4], which is a web-based watermark benchmark platform. The online service[1] provides the user the possibility to embed, attack and retrieve information by using selected watermark and attack algorithms for still images.

[1] The system is located at http://www.datahiding.org for images and http://audio-wet.ecn.purdue.edu or http://audio-wet.cs.uni-magdeburg.de for audio

Security, Steganography, and Watermarking of Multimedia Contents VII, edited by
Edward J. Delp III, Ping Wah Wong, Proc. of SPIE-IS&T Electronic Imaging,
SPIE Vol. 5681 © 2005 SPIE and IS&T · 0277-786X/05/$15

The WET server contains the web server software, an image database, and image processing software. The web server presents the front end and user interface for selecting images to watermark, watermarking techniques to evaluate, embedding and detection parameters, and attacks. The web interface is also used to present the evaluation results to a user. The image database stores the images for subsequent watermarking and processing. The database also contains image characteristics and metadata, which allows the user to select subsets of the images for evaluation. The image processing software performs the watermark embedding, watermark detection, and evaluation functions. No software is required by the user to interact with the WET server except a web browser capable of displaying images and JavaScript. The current implementation of WET uses all open-source software, including the Apache [11] web server with PHP [12], MySQL [13] for the image database, and GIMP [14] for image processing.

The architecture of the WET system is based on an extensible and flexible modular design. Modules are used for implementing watermark embedders, detectors, and attacks. Evaluation is also performed by the use of modules. Watermarking techniques currently implemented in WET include [15][16][17][18][19][20][21]. Additional modules for watermarking, attacks, and evaluation are under development. WET is also used to study the theory of watermark evaluation and to develop methodologies for benchmarking.

2 StirMark for Audio

In this section, we introduce StirMark for audio. Furthermore, we present the profiles and we discuss the assignment of profiles as well as the definition of profile parameters. The StirMark Benchmark for Audio (SMBA) is a process for evaluating digital audio watermarks. The general usage of SMBA is as follows: First, a digital robust audio watermark is embedded into the original audio file by using a watermark embed algorithm. Then the watermarked audio file is evaluated by using SMBA. This process works with many different attacks, which can be configured by using attack parameters. The selected SMBA attack modifies the audio signal a little bit with the goal to destroy or weak the embedded watermark information. Other attacks try to destroy or weak the synchronization of the watermark to impede to detection of watermark synchronization. After attack process, the watermark detector tries to detect and retrieve the watermark and gives results about the watermark information. So the SMBA is just only the attack process in the whole evaluation or benchmark process of digital watermarks.
The simplest way to attack a digital watermark is a brute force attack by using every possible attack against the watermark. For each attack, StirMark has default attack parameters, which can be used to evaluate the digital audio watermark very quickly. It is also possible to change and optimize the attack parameters to improve the attack strength or attack transparency. Each attack is a single evaluation process and the user knows where the watermarking algorithm is weak or with which attack the watermark can be broken. These single attacks are atomic signal modification processes.
This scenario is also called “single attack process” [5]. The watermarked audio file undergoes many attacks and produced for each attack a separate audio file. Each of these audio files is only modified by a single attack (e.g. add noise, change the pitch, change the amplitude or cut samples). This is useful to find a plainly weakness of a watermark algorithm. By using this attack method, the user has the problem, that many different attacked audio files are created. The watermark algorithm has to try to retrieve the watermark information to verify the robustness against the attack.

Another attack mode is called “profile attack” [3][5] and runs more than one attack in serial order against the digital watermark.
An evaluation profile is an ordered sequence of processes that may be applied to a signal, as shown in Figure 1. Each of the individual processes in the profile is defined by its own set of parameters. While a profile may seem to be merely an attack or process macro, profiles serve a very useful purpose in benchmarking. Profiles allow the evaluation system to model or simulate scenarios of interest to particular applications. An evaluation profile may be defined in terms of other (existing) profiles, which allow a complex process or attack to be modeled as a sequence of previously-defined (or elementary) processes (for example the DA/AD conversion)

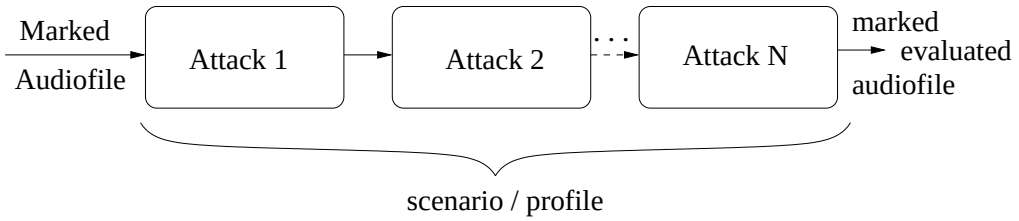

Figure 1: Evaluation profiles [5]

The challenge is to define the profiles and their parameters in a manner that is both flexible and interesting for evaluating the performance of audio watermarking in specific applications.

.

2.1 Profiles and StirMark for Audio

In this section we want do introduce as first the new defined profiles *packet loss* and *watermark detection* and we describe what the profiles itself can do.

In [3] are defined main profiles, which differ among low and high quality as well as robustness and fragility, and sub-profiles, which describe the real world application, for benchmarking systems and we want to use these profiles to enhance them by defining new profiles and insert them into the existing WET service. The profiles are used to evaluate a digital watermark not just only with a single attack. Profile attacks are an attack scenario, which comes from the real world (like play music or streaming). These profiles are composite with many single attacks. Some profiles can be implemented by using StirMark for audio itself and other profiles can be simulated by using other tools.

For example: The profile *lossy compression rates* can be implemented by using different existing lossy compression encoders. We prefer to use the widespread MP3 and the completely royalty free encoder OGG. In the follow chapter 3 we describe how we implement the profile.

Another real world application is the transmission of audio content via Internet and we simulate it with a new profile *Packet Loss*. There are many applications, where audio data are split into IP packets, transmitted and routed over public networks like IP-Telephony, Internet radio or something like this. Depending on the Internet protocol a packet loss can be detected and remedied. However, if a packet loss is detected it will take some time to reorder the lost packet. For an application like IP-Telephony or Internet radio it is not recommended to reorder the lost packet. The reason is the order of received packets and the transmission time. The reordered packet will receive the receiver too late and to play the audio data will confuse the listener. To understand the packet loss, the followed description will help:

An audio data stream is transmitted over Internet by using IP packets. The sender splits the audio data and packs them into the packets *A B C D E F G*. If no packet loss occurs, the receiver gets the packets in the same order *A B C D E F G*, reconstructs the audio data and plays it. If for example the packet *C* is lost and not reordered, the audio data are only *A B D E F G*. The part *C* is missing and a depending on the used audio codec, the listener can hear an artifact. If the receiver reorder the packet, by re-request the missing packet, the receive order can be *A B D E F C G*. Now the problem is, that the audio data from to late received packet *C* does not fit to part *F* and *G*. In this case, a listener will hear an artifact between *B* and *D* and a not matched sound or voice between *F* and *G*. This is the reason, why real time audio transmission applications do not reorder lost packets. The profile *Packet Loss* describes the introduced scenario and in chapter 3 we discuss the implementation of them.

Another new profile is *Watermark Detection*. The idea behind this profile comes from the steganalysis. By using the profile *Watermark Detection*, the result will not be a new attacked audio file. Rather the profile gives a probability of the existence of an embedded digital watermark. Here we want to use statistical analysis to detect hidden information [6].

2.2 Assignment of Profiles

We have designed a set of profiles that we believe encompasses many of the performance issues that are important in audio watermarking applications. Figure 2 shows our profiles. The main and all-inclusive (robust and/or fragile and/or invertible) watermark is a hybrid watermark. Here are combined different types of existing watermarks. A typical application scenario is to identify a copyright owner and to detect manipulation of the content. Such scenario includes mostly the combination of a robust and fragile digital watermark. A hybrid watermark has special parameters and each parameter can be evaluated by the corresponding profile. The lines between the profiles show

the dependencies of the sub profiles (for e.g. *Coalition Resistance* or *Degree of Fragility*) to the corresponding main profiles (*High* and *Low Quality Robust* and *Fragile*).

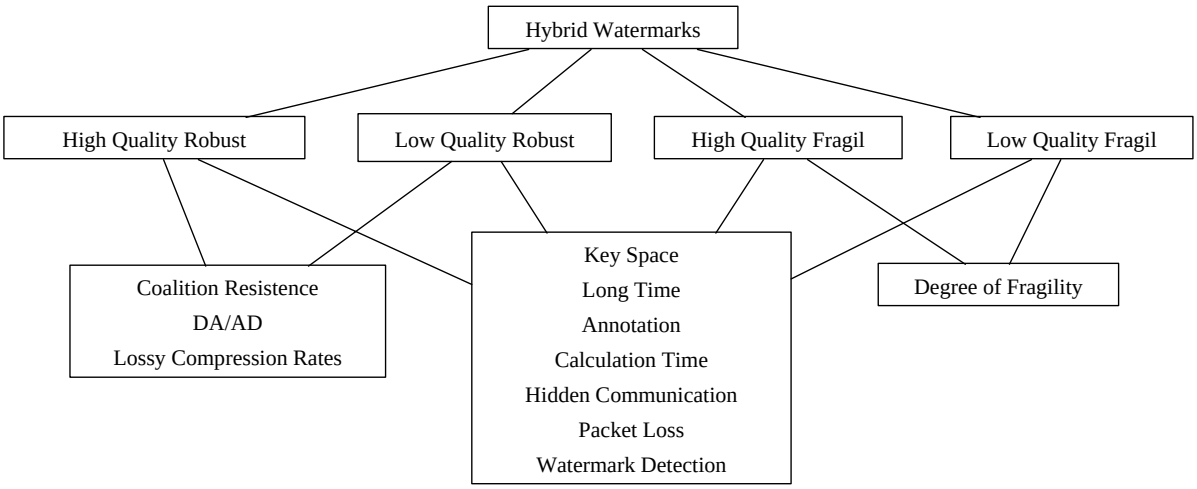

Figure 2: Assignments of Profiles

The main profiles and sub profiles are defined in [3]. The assignment of the profiles is important because of the classification and categorization of sub-profiles to specify the general quality level as well as the type (robust or fragile) evaluation process. This means, that some sub profiles are only useable for some (corresponding) main profiles. One example is the profile *Lossy compression rates*. This profile is a sub profile from the main profiles *High Quality Robust* and *Low Quality Robust*. Depending on the data rate of the lossy compression encoder, the main profile is selected.
Other sub profiles (for example *Calculation time*) are useable with all main profiles. The concrete definitions are shown in subsection 2.3 where the profile parameters are introduced.

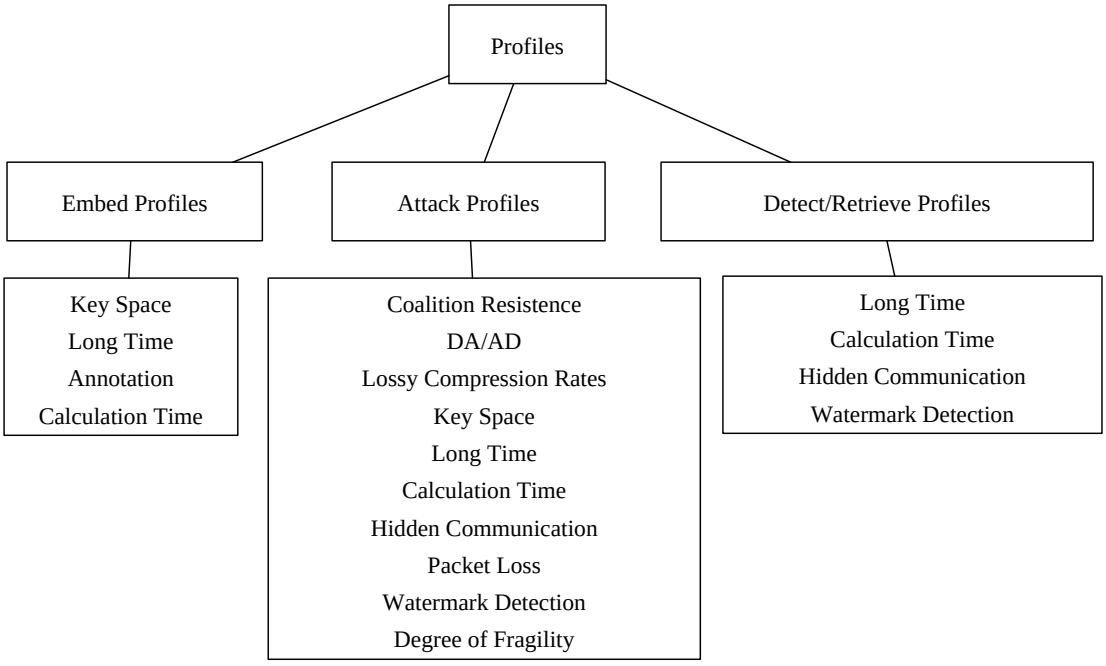

Figure 3: Profile classes

Another important assignment is the assignment of profiles to the type of profile class (*Embed*, *Attack* or *Detect/Retrieve Profiles*). It means, that the type of main usage of embedded information can be *embed*, *retrieve* or *detect* and *attack*. To understand the main usages, the followed itemization introduces the three main usages:

- The main usage profile class *Embed* is associated with the embedding process of a hidden information by using a watermark or steganographic algorithm.
- The main usage profile class *Attack* has all sup-profiles, which specify a sub-profile which attacks watermarked signal.
- The main usage profile class *Detect/Retrieve* is associated with the detection or retrieving process of a hidden information.

The defined sub profiles can be assignment to one, two or three profile classes. The Figure 3 shows the assignment of sub-profiles. There are sub-profiles (for example Annotation) which is just only an embed profile. Other sub-profiles (for example Long Time) are assignment to all three profile classes.

2.3 Definition of profile parameter

In this subsection, we want do introduce the profile definition and profile parameters by using a profile description language. Firstly, we define the variables. In the second step, we use the defined variables to introduce the profiles itself. The last step introduces selected profiles and discusses these.

We use the following notation and definitions for describing profiles and profile parameters:

A – set of all attacks
A_i – specific attack from A
P – set of all profiles
$P_{j\text{-}i}$ – specific profile from P, j={*E, A, D*}, E=Embed, A=Attack, D=Detect, i – specifies the profiles
x – attack or profile parameters
i – specifies the attack or profile by using the name
j – there are only three letters possible: *E* for embed profile, *A* for attack profile and *D* for detect/retrieve profile

If a profile P is specified, the assignment of main usage (embed, attack, detect/retrieve) is specified using a capital latter followed by a "-" (minus). After the "-" comes the name of the profile. If the profile name has more than one word, the words are separated by a "_" (underline). If it is not important which main usage is needed, than the specific letters can be separated by a "/" (slash). To improve the understandings, we introduce two examples for the attack *Invert* and the profile *Lossy Compression Rates*:

$A_{Invert}(paramlist)$ – the single attack invert
$P_{A\text{-}Lossy\_Compression\_\mathrm{Rates}}(paramlist)$ – lossy compression profile with the main usage: attack
$P_{E/A/D\text{-}Long\_Time}(paramlist)$ – sub-profile Long Time used in all three main usages

The value *paramlist* indicates the list of parameters, whose definitions are profile-dependent. Each parameter in the parameter list is separated by the symbol ||. The first two parameters will always be the input signal and output signal, and the remaining parameters depend on the profile, as shown below:

paramlist = (*in-signal* || *out-signal* || *more params*)

in-signal S' - Type of media input signal. Can be a file (pcm, wav, mp3, ogg, …) or a signal stream from a special source like microphone.
out-signal - Type of media output signal. Can be a file (pcm, wav, mp3, ogg, …) or a signal stream to a special destination like loud speaker.
more params - If the attack or profile needs additional parameters, the parameters are defined here. For each profile, these parameters differ and will be described below for some typical profiles. If more than one parameter is needed, the parameters are concatenated.

Followed, we define the formal description of the defined profiles. After them, we discuss the value *parameter* from x for selected profiles.
The followed Table 1 shows the profile name and the formal description of the profile itself.

Table 1: Formalization of profiles

Profile name	Formal description
Annotation	$P_{E\text{-}Annotation}(x)$
Calculation Time	$P_{E/A/D\text{-}Calculation\_Time}(x)$
Coalition Resistance	$P_{A\text{-}Coalition\_Resistance}(x)$
DA/AD	$P_{A\text{-}DA/AD}(x)$
Degree of Fragility	$P_{A\text{-}Degree\_of\_Fragility}(x)$
Hidden Communication	$P_{A/D\text{-}Hidden\_Communication}(x)$
Key Space	$P_{E/A\text{-}Key\_Space}(x)$
Long Time	$P_{E/A/D\text{-}Long\_Time}(x)$
Lossy Compression Rates	$P_{A\text{-}Lossy\_Compression\_}\text{Rates}(x)$
Packet Loss	$P_{A\text{-}Packet\_Loss}(x)$
Watermark Detection	$P_{A/D\text{-}Watermark\_Detection}(x)$

Followed, we want to introduce and discuss different profiles as example. The selected profiles *Lossy Compression Rates, DA/AD, Hidden Communication, Watermark Detection* and *Degree of Fragility* are described in their parameters.

- *Lossy Compression Rates:*
  This profile *Lossy Compression Rates* is needed to simulate different lossy compression rates for audio signal and simulate them.

  $P_{A\text{-}Lossy\_Compression\_}\text{Rates}(in\text{-}signal \,||\, out\text{-}signal \,||\, parameters)$

  $parameters = (algorithm \,||\, data\ rate \,||\, option)$

  algorithm - Defines the lossy compression algorithms to compute the lossy compression. Typical compressors are mp3, ogg, wma or vqf.
  data rate - Specifies the data rate in *kbit/s* which is used to encode the *in-signal*.
  option - Optional the user can set specific parameters which are depend on the lossy compression algorithms. Typical optional parameters can be static or variable data rate or low- or high pass frequency filtering.

  This sub-profile is associated to the main profiles *High*- and *Low Quality Robust*. The classifier to select the main profile is the *data rate* parameter. The data rate can be 8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 144, 160, 192, 224, 256 and 320 kbps. We suggest the data rate of 128 kbps as threshold between the both main profiles. A data rate less that 128 kbps is associated to the main profile *Low Quality Robust*. If the data rate is equal or higher than 128 kbps, then the associated main profile is *High Quality Robust*. We know, that a lot of mp3 users suggest using a data rate equal or more than 192 kbps to get a high mp3 quality. One point is that this data rate is a specific value for mp3 compression and we think that the data rate of 128 kbps is a good threshold.

- *DA/AD:*
  This profile *DA/AD* is needed to simulate a digital-analogue and analogue-digital conversion of audio signals.

  $P_{A\text{-}DA/AD}\ (in\text{-}signal \,||\, out\text{-}signal \,||\, parameters)$

  $parameters = (A_{i1}(x_1) \,||\, A_{i2}(x_2) \,||\, A_{i3}(x_3) \,||\, \ldots \,||\, A_{in}(x_n))$

  $A_{i1}(x_1)$ - $A_{in}(x_n)$ - Are single attacks, which have to run in the order $i=1\ldots n,\ i{+}{+}$ with the using of single attack parameters $x_1 \ldots x_n$.

  To specify the attack A_i and the appendant parameters x_i is not completely clear. The reason is the used hardware for playing and recording. By using different loud speaker and/or different microphones the

attack parameters x_i have to change. If there is just only a cable between the DA/AD converter, the number of attacks decrease and the attack parameters are changed.
At this time we did not specify the exact A_i and x_i to simulate the DA/AD conversion.

- *Hidden Communication, Watermark Detection:*
  The Profile *Hidden Communication* is needed for detection of embedded information in audio signals by using a steganographic technique. The profile *Watermark Detection* is for detection of embedded information in audio signals by using a watermarking algorithm.

  $P_{A/D\text{-}Hidden\_Communication}$(*in-signal* || *out-signal* || *parameters*)
  $P_{A/D\text{-}Watermark\_Detetcion}$(*in-signal* || *out-signal* || *parameters*)

  in-signal - Type of media input signal. Can be a file (pcm, wav, mp3, ogg, …) or a signal stream from a special source like microphone.
  out-signal - likelihood of embedded information or visualization of certain statistical analyzes

  parameters = (*algorithm* || *option*)

  algorithm - Kind of steganalyze algorithm to compute the likelihood of embedded information.
  option - Optional the user can set specific parameters which are depend on the steganalysis algorithm. Typical optional parameters can be a special part of the audio signal or thresholds for skipping of irrelevant sample values.

  This sub-profile is associated to the main profiles *High*- and *Low Quality Robust* as well as *High*- and *Low Quality Fragile*. The reason is that this profile does not touch the characteristic of the audio signal and it does not care about the quality. It is unimportant if the input audio signal has high or low quality or if there is embedded robust or fragile information.

- *Degree of Fragility:*
  The Profile *Degree of Fragility* is needed to simulate manipulation on the audio signal or audio content. This profile can change from one bit (for bit fragile watermarking) up to audio content modification (for content fragile watermarking). Depending on the fragile audio watermark algorithm, the manipulation can be or should be detected.

  $P_{A\text{-}Degree\_of\_Fragility}$(*in-signal* || *out-signal* || *parameters*)

  parameters = (*modification* || *option*)

  modification - This parameter specifies the type of manipulation of the audio signal. It can be just one bit or a special cut of specified audio content. This parameter selects the strength of degree of the audio signal.
  option - Depending on the degree level, the modification needs additional parameters to specify the modification. Typical examples are: if the bit change is selected, the number of bits per second is needed too. Another example is the selection of a degree level of content fragility, the strength of modification.

  This sub-profile is associated to the main profiles *High*- and *Low Quality Fragile*. It specifies the manipulation of the audio signal in different levels. The simplest manipulation is the fall over of a bit to evaluate bit fragile watermark algorithms. Heaviest is the modification of the signal with and without changing of the content itself. If the audio signal is changed or transformed and the content is not changed, then a content fragile watermark should be alive. But if the content is changed (for example a part is cut off), then the content fragile watermark should detect this manipulation.

2.4 Parameter range

In this subsection, we want to discuss the parameters for the attacks to build the profiles itself as well as the impact to the watermark properties (capacity, transparency, robustness/fragility, complexity, security, verification, invertibility) [5]. To create the profiles, different tools (for example: en-, decoder and SMBA) are needed.

The followed Table 2 shows the sub-profiles. In the next columns, the used tool and function is listed as well as the parameter of the function/attacks and a short description.

Table 2: Parameters and tools

Sub-Profile	Tool	Function/Attack(s)	Parameter	What doing
$P_{E\text{-}Annotation}(x)$	Embed-Algorithm	Embed	Strength, Capacity,	Measure the capacity of an embed algorithm
$P_{E/A/D\text{-}Calculation\_Time}(x)$	Time measure tools	All functions (embed, retrieve, detect, attack)	Complexity	Measure the used time to do the selected function.
$P_{A\text{-}Coalition\_Resistance}(x)$	Not specified at this time. But we will implement it later.	Attack	Strength	Measure the resistance against coalitions.
$P_{A\text{-}DA/AD}(x)$	SMBA	Attack (Add_Noise, Amplify, High-, Lowpass, Distorion,…)	Strength, Order	Simulates a DA/AD conversion to measure the robustness
$P_{A\text{-}Degree\_of\_Fragility}(x)$	SMBA	Attack (not implemented)	Strength, fragility	Measure the fragility level
$P_{A/D\text{-}Hidden\_Communication}(x)$	Steganalyzer for MP3 and PCM	Detect	transparency	Detects hidden information by using different statistical analysis techniques.
$P_{E/A\text{-}Key\_Space}(x)$	Not specified at this time. But we will implement it later.	Attack	Unknown	Reducing the key space and trying a brute force attack to read the hidden information.
$P_{E/A/D\text{-}Long\_Time}(x)$	No special tool	Embed, attack, detect, retrieve	Length of audio signal	Measure the behavior of the algorithm.
$P_{A\text{-}Lossy\_Compression\_}\text{Rates}(x)$	En- and Decoder	Attack	Data rate or quality level	Measure the robustness by a specified data rate.
$P_{A\text{-}Packet\_Loss}(x)$	SMBA	Attack (Cut_Sample)	Number of packet loss: robustness, fragility	Measure the robustness or fragility against loss of signal information.
$P_{A/D\text{-}Watermark\_Detection}(x)$	Steganalyzer for MP3 and PCM, SMBA	Detect	Transparency	Detects hidden information by using different statistical or analysis techniques.

To get the assignment of the sub-profiles to the main profiles, we introduce in the following Table 3 the dependences. The word *independent* means, that the sub-profile does not care about the assignment to the main profile. The reason is that the sub profile does not have an impact about the quality level, robustness or fragility property of the main profile. For example: $P_{E/A/D\text{-}Calculation\_Time}(x)$ is needed to measure the complexity of an algorithm (embed, attack or retrieve). Neither the quality level nor the robustness or fragility does effect of this profile.

Table 3: Assignment of sub- and main-profiles

Sub-Profile	Parameter and assignment main profile
$P_{E\text{-}Annotation}(x)$	Independent
$P_{E/A/D\text{-}Calculation\_Time}(x)$	Independent
$P_{A\text{-}Coalition\_Resistance}(x)$	Just only *High*- and *Low Quality Robust:* Depending on the coalition algorithms, this sub-profile is *High*- or *Low Quality*.

	After implementing of this profile, we can specify the exactly parameters.
$P_{A\text{-}DA/AD}(x)$	*High Quality Robust*: if the DA/AD process does not have a "strength" impact of the audio signal *Low Quality Robust*: if the DA/AD conversion is measure or hearable We will specify the needed parameters in the future. Maybe the objective difference grade (ODG) is useful.
$P_{A\text{-}Degree\_of\_Fragility}(x)$	
$P_{A/D\text{-}Hidden\_Communication}(x)$	Independent
$P_{E/A\text{-}Key\_Space}(x)$	Independent
$P_{E/A/D\text{-}Long\_Time}(x)$	Independent
$P_{A\text{-}Lossy\_Compression\_Rates}(x)$	*High Quality Robust*: data rate >= 128 kbits/s *Low Quality Robust*: data rate < 128 kbit/s
$P_{A\text{-}Packet\_Loss}(x)$	*High Quality Robust:* if packet loss can be "repaired" (ECC) *Low Quality Robust:* if packet loss has an impact of the audio signal *High Quality Fragile:* if packet loss can be "repaired" (ECC) and the fragility is warranted *Low Quality Fragile:* if packet loss has an impact of the audio signal and the fragility is warranted
$P_{A/D\text{-}Watermark\_Detection}(x)$	Independent

Furthermore, our idea is to assign for the used tool (see in Table 3) a set of possible parameter for the function/attack. To give an example we discuss our approach for two profiles, the $P_{A\text{-}Lossy\_Compression\_Rates}(x)$ and the $P_{A\text{-}Degree\_of\_Fragility}(x)$.

The following Table 4 shows the exactly *parameters* for the profile $P_{A\text{-}Lossy\_Compression\_Rates}$(*in-signal* || *out-signal* || *parameters*) belonging to the main profiles. As described above: *parameters* = (*algorithm* || *data rate* || *option*). At this time, WET for audio has two lossy compression algorithms. The first is MP3 and the second is OGG. As encoder for MP3 we use *Lame* [7] and as decoder *mpg123* [8]. To compute the OGG lossy compression, we use *oggenc* as encoder and *ogg123* as decoder. Both programs are part of *vorbis-tools* [9]. Future work is to attach other lossy compression algorithms (WMA, VQF) to the audio WET system depending on the licence model.

Table 4: Attack parameter for sub-profile $P_{A\text{-}Lossy\_Compression\_Rates}(x)$

Algorithm	Data rate or quality level	Main profile
MP3 (lame encoder)	< 128 kbit/s	*Low Quality Robust*
MP3 (lame encoder)	>= 128 kbit/s	*High Quality Robust*
OGG (Vorbis encoder)	< quality lever of 3	*Low Quality Robust*
OGG (Vorbis encoder)	>= quality level of 3	*High Quality Robust*

Our second example is the profile $P_{A\text{-}Degree\_of\_Fragility}(x)$. To use this profile, it is different what kind of fragile watermark algorithm is used. We distinguish between *bit fragile* and *content fragile* watermarking algorithms. By using a *bit fragile* watermark algorithm, the watermark should be broken if only 1 bit of the whole signal is changed. Here we plan to implement a new attack in SMBA which changes one bit randomly to simulate the bit overturn attack. By using a *content fragile* watermark, the watermark should be alive after attacks, which do not change the audio content itself. The followed Table 5 shows the attacks in SMBA and the impact of a *content fragile* watermark. Depending on the strength of the attack, the main profile is High Quality or Low Quality.
The second column (impact) has a *yes* or *no*. The meaning behind a *yes* is that this attack can destroy a *content fragile* watermark depending on the attack parameters. A *no* means that this attack should not have any impact on the content fragile watermark. But if the attack parameter is to strong, then the watermark is destroyed and the transparency of the attack is not warranted.

Table 5: Attack parameter for sub-profile $P_{A\text{-}Degree\_of\_Fragility}(x)$

Attack in SMBA	Impact	Attack parameter	Comment
Add_Brumm	No	Strength: 0…2000 Frequency=55Hz	
AddDynNoise	No	Strength: 0…100	
AddFFTNoise	No	FFTSize: 1024, 2048 Strength: 0…5000	
AddNoise	No	Strength: 0…70	
AddSinus	No	Strength: 0…100 Frequency: 0.22050	
Amplify	No	Factor: 50…200	
BassBoost	No	Threshold: 100Hz BosstDB: <3dB	
Compressor	No	Threshold: -6.1 CompressFactor: 2.1	
CopySample	Yes	Period: <88200 CopyDistance: < 44100 CopyCounter: >= 1	The CopyCounter value has to be determined for each algorithm.
CutSample	Yes	Remove: < 88200 RemoveNumber: >=1	The RemoveNumber value has to be determined for each algorithm.
Echo	Yes/No	Period: >0	
Exchange	No	None	
ExtraStereo	Yes/No	Strength: 1…100	The strength has to be determined for each algorithm.
FFT_HLPassQuick	No	FFTSize: 1024, 2048 HP_Freq: 100Hz LP_Freq: 16000Hz	
FFT_Invert	No	FFTSize: 1024, 2048	
FFT_RealReverse	No	FFTSize: 1024, 2048	
FFT_Stat1	No	FFTSize: 1024, 2048	
FlippSample	Yes	Period: 88200 FlippCount: <6000 FlippDistance: <2000	The FlippCount and FlippDistance value has to be determined for each algorithm.
Invert	No	None	
LSBZero	No	None	
NoiseMax	No	Mask: 23 Length: 1365 Value: < 200	
Normalize	No	None	
Nothing	No	None	
Pitchscale	No	FFTSize: 1024, 2048 ScaleFactor: 0.9…1.1 Windowing: 32	
RC-HighPass	No	HP_Freq: 100Hz	
RC-LowPass	No	LP_Freq: 16000Hz	
Smooth	No	None	
Smooth2	No	None	
Stat1	No	None	

Stat2	No	None	
TimeStrech	No	TempoFactor: 0.9…1.1	
VoiceRemove	Yes	None	
ZeroCross	No	Threshold: 0…2000	
ZeroLength	Yes	Length: >1	The length has to be determined for each algorithm.
Zerolength2	Yes	Length: >1	The length has to be determined for each algorithm.
ZeroRemove	Yes	None	

3 Integration

To make the profiles useable for both the WET system and SMBA, it is necessary to describe the profiles in a unique, defined language. In this language it should be possible to describe the audio and image profiles, multiple attacks on multiple signals and all needed parameters. We decided to use XML [10] to describe the profiles. The profile description language (or schema) is a work-in-progress. Figure 4 shows a conceptual example of a profile description, but the final version of the profile description language may be much different in structure than this example.

```
<profile name="Profile_DA/AD">
 <attack>
        <add_noise>
            <parameter>
                <strength> 100 </strength>
            </parameter>
        </add_noise>
        <fft_hlpassquick>
            <parameter>
                <fft_size> 1024 </fft_size>
                <hp_freq> 100 </hp_freq>
                <lp_frq> 17000 </lp_freq>
            </parameter>
        </fft_hlpassquick>
        …
 </attack>
 …
</profile>
```

Figure 4: Conceptual example of profile description

4 Conclusion

We have described attack profiles for audio and the integration concept into WET in this paper. In subsection 2.2, we introduced the assignment and the main usage of all defined profiles. The formal profile definition and the associated profile parameters are introduced in subsection 2.3. We selected 4 typical profiles (Lossy Compression Rates, DA/AD, Hidden Communication and Degree of Fragility) and discussed them in more detail. Furthermore, the follow subsection discussed the value of parameter ranges and the assignment of sub- main profiles. To integrate the profiles in the existing WET system, we selected XML as description language and we introduced it briefly. The next steps includes the implementation of the profiles itself and the detailed definition of the XML language which is needed to describe and store the selected embedding, attacking and detecting jobs from the client computer.

Acknowledgments

Effort 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.

Furthermore basic ideas presented here come from the previous works done in StirMark and we would like to thank all involved persons for their stimulating discussions and their work. Furthermore we would like to thank our student Tobias Scheidat for his help in structuring and implementing the PHP scripts and Claus Vielhauer, Thomas Vogel and Danny Hesse for there fundamental discussions.

References

[1] Macq, B.; Dittmann, J.; Delp, E.J., *Benchmarking of Image Watermarking Algorithms for Digital Rights Management*, Proceedings of the IEEE, Journal, pp. 971- 984, 2004
[2] StirMark for Audio, http://amsl-smb.cs.uni-magdeburg.de
[3] Andreas Lang, Jana Dittmann, *StirMark and profiles: from high end up to preview scenarios*, Reviewed Paper, IFIP/GI Workshop on Virtual Goods, Ilmenau (Germany), 28-29 May 2004, online publication available from http://virtualgoods.tu-ilmenau.de/2004/program.html
[4] Hyung Cook Kim, Hakeem Ogunleye, Oriol Guitart, Edward J. Delp, *The Watermark Evaluation Testbed (WET),* San Jose, CA, USA Bellingham, Washington, USA, SPIE 2004, vol. 5306
[5] Jana Dittmann, Martin Steinebach, Sascha Zmudzinsky, Andreas Lang, *Advanced audio watermarking benchmarking*, San Jose, CA, USA Bellingham, Washington, USA, SPIE 2004, vol. 5306
[6] J. Dittmann, D.Hesse, R.Hillert, *Steganography and Steganalysis in Voice over IP Scenarios - Operational Aspects and First Experiences with a New Steganalysis Tool Set*, to appear in SPIE 2005, San Jose
[7] LAME Ain't an MP3 Encoder, http://lame.sourceforge.net/, 2004
[8] W. Richard Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, 1994
[8] mpg123, Fast MP3 Player for Linux and UNIX systems, http://www.mpg123.de, 2004
[9] xiph.org: Ogg Vorbis, http://www.xiph.org/ogg/vorbis/, 2004
[10] The XML C parser and toolkit of Gnome, http://xmlsoft.org/, 2004
[11] Apache Software Foundation, *The Apache HTTP Server Project*, http://httpd.apache.org/, 2004
[12] Hypertext Preprocessor, http://www.php.org, 2004
[13] MySQL Database, http://www.mysql.com/, 2004
[14] The GNU Image Manipulation Program, http://www.gimp.org/, 2004
[15] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, *Secure spread spectrum watermarking for multimedia*, IEEE Transactions on Image Processing, vol. 6, no. 12, pp. 1673-1687, 1997.
[16] R. Dugad, K. Ratakonda, and N. Ahuja, A new wavelet-based scheme for watermarking images, Proceedings of the IEEE International Conference on Image Processing, Chicago, IL, Oct. 1998.
[17] E. T. Lin, C. I. Podilchuk, and E. J. Delp, Detection of image alterations using semi-fragile watermarks, Proceedings of the SPIE/IS&T Conference on Security and Watermarking of Multimedia Contents, P. W. Wong and E. J. Delp, Eds., vol. 3971, January 2000, pp. 152-163.
[18] B. Chen and G. W. Wornell, Quantization index modulation: A class of provably good methods for digital watermarking and information embedding, IEEE Transactions on Information Theory, vol. 47, no. 4, pp. 1423-1443, May 2001.
[19] Y. Liu, B. Ni, X. Feng, and E. J. Delp, LOT-based adaptive image watermarking, Proceedings of the SPIE/IS&T Conference on Security and Watermarking of Multimedia Contents, P. W. Wong and E. J. Delp, Eds., vol. 5306, San Jose, CA, January 2004.
[20] E. T. Lin and E. J. Delp, Locktography: Technical description, internal Purdue document.
[21] A. Piva, M. Barni, F. Bartolini, and V. Cappellini, DCT-based watermark recovering with restoring to the uncorrupted original image, Proceedings of the IEEE International Conference on Image Processing, vol. 3, 1997, pp. 520-523.