A Computational Model for Watermark Robustness

André Adelsbach[1], Stefan Katzenbeisser[2], and Ahmad-Reza Sadeghi[3]

[1] Horst Görtz Institute for IT Security, Ruhr-Universität Bochum
andre.adelsbach@nds.rub.de
[2] Institut für Informatik, TU München
katzenbe@in.tum.de
[3] Horst Görtz Institute for IT Security, Ruhr-Universität Bochum
sadeghi@crypto.rub.de

Abstract. Multimedia security schemes often combine cryptographic schemes with information hiding techniques such as steganography or watermarking. Example applications are dispute resolving, proof of ownership, (asymmetric/anonymous) fingerprinting and zero-knowledge watermark detection. The need for formal security definitions of watermarking schemes is manifold, whereby the core need is to provide suitable abstractions to construct, analyse and prove the security of applications on top of watermarking schemes. Although there exist formal models and definitions for information-theoretic and computational security of cryptographic and steganographic schemes, they cannot simply be adapted to watermarking schemes due to the fundamental differences among these approaches. Moreover, the existing formal definitions for watermark security still suffer from conceptual deficiencies.

In this paper we make the first essential steps towards an appropriate formal definition of watermark robustness, the core security property of watermarking schemes: We point out and discuss the shortcomings of the existing proposals and present a formal framework and corresponding definitions that cover those subtle aspects not considered in the existing literature. Our definitions provide suitable abstractions that are compatible to cryptographic definitions allowing security proofs of composed schemes.

1 Motivation

Multimedia applications deploy various cryptographic and watermarking techniques to maintain security. Typical application scenarios are dispute resolving, proof of authorship and asymmetric and anonymous fingerprinting.

In this context, the security analysis and security proofs for the resulting composed schemes require a suitable formal framework and reasonable security definitions. Modern cryptography already uses established formal models and definitions for information-theoretic and computational security. Inspired by cryptographic methodology similar approaches have been proposed for

J. Camenisch et al. (Eds.): IH 2006, LNCS 4437, pp. 145–160, 2007.

steganography [1,2,3,4]. In contrast, less investigation has been done with this regard for watermarking schemes, and the existing approaches do not cover the subtle aspects essential for reasonable formal security definitions, analysis and abstraction of watermarking schemes.

The need for formal definitions of watermarking schemes, and their most notable properties, such as robustness, false-positive and false-negative probabilities, is manifold: first, one requires formal definitions as suitable abstractions to build, analyse and prove the security of applications on top of watermarking schemes. Second, one requires suitable formal definitions to prove the robustness of watermarking schemes. Furthermore, such definitions provide valuable guidance and basis in the development of provably robust watermarking schemes.

One should note that steganography, although likewise watermarking a means for information hiding, differs from watermarking with respect to various aspects. The most important difference concerns their requirements: In steganography there is a strong hiding requirement, stating that an adversary cannot even detect the *presence* of some stego-message in stego-data. In watermarking, however, one usually does want to prevent watermarks to be detectable by an adversary.[1] Instead, the challenging core property, distinguishing watermarking schemes from other cryptographic or data-hiding primitives, is the robustness property, which guarantees that a watermark cannot be removed without significantly distorting the stego-data and making it useless.[2] Due to the fundamental difference between steganography and digital watermarking, one cannot simply adapt recent definitions of steganographic security [3,4].

In this paper, we point out and discuss the shortcomings of the existing proposals for watermark security definitions as well as the subtle aspects/parameters that these proposals do not cover. In fact, our review shows that even the meaning of watermark security is still not well understood, mainly because many authors do not focus on the main, distinguishing security property of watermarking schemes, which cannot be achieved by applying complementary cryptographic measures: *robustness*. We propose formal (and intuitive) definitions for watermarking schemes, including robustness, that (i) incorporate these aspects/parameters and (ii) can be used as a suitable abstraction for security proofs of composed schemes in the context of various applications.

2 Related Work

In recent years, there has been a remarkable body of literature on definitions for robustness and security of watermarking schemes. Most of the existing proposals

[1] One may require watermarking schemes to provide an optional secrecy property, requiring that adversary cannot obtain any information about the concrete watermark embedded in the stego-data. This requirement is very different and much easier to achieve (e.g., by using standard encryption schemes) than the strong hiding property which is at heart of steganographic systems.

[2] Note, that we do not consider fragile watermarking schemes, because fragility can be achieved quite easily, using cryptographic primitives.

distinguish between the security and robustness of a watermark. In this context robustness concerns the amount of information on watermark that is revealed to an adversary, whereas the security often concerns the information revealed on secret embedding key. The corresponding definitions are based on information-theoretical or cryptographic methodologies.

Mittelholzer [2] proposed the first formal model, which defines *information-theoretic robustness* in terms of mutual information[3]: a robust watermarking scheme is defined to maximise the mutual information $I(WM; W''|K^{\mathtt{det}})$ between the watermark WM and the distorted stego-data W'', when given the detection key $K^{\mathtt{det}}$. The maximum is defined over all allowed channels (adversaries), transforming watermarked data W' into distorted data W''.

Kalker [5] introduced reasonable but informal definitions of watermark robustness and watermark security:[4] watermark robustness is defined as the property that the capacity of the watermarking channel degrades as a smooth function of the degradation of the stego-data. Security is defined as the inability of an adversary to remove[5], detect (or estimate), write or modify any bit of the watermark. The notion of "security" is very broad and, therefore, too strong for most applications of watermarking schemes.

Barni et al. [6] proposed a general security framework for watermarking systems, where they measure security by quantifying the information on the secret watermarking key that is leaked through stego-data the adversary can observe. The authors define security in terms of a two party game between a correct party and the adversary. The rules of the game determine the a-priori information given to the adversary and which he may use to win the game, i.e., break the respective security property of the watermarking scheme. In principle, this is a common approach in cryptography when defining security properties of cryptographic schemes. However, in [6] the authors distinguish between *fair* and *unfair* adversaries: according to the games' rules fair adversaries only use the a-priori information, whereas unfair adversaries try to gain secret information and take advantage of this knowledge. The distinction between fair and unfair adversaries is uncommon and restricts the adversary's strategies covered by the definitions and, thereby, weakens the definition significantly. For instance, the definition does not cover adversaries who exploit weaknesses of the watermarking scheme to get information about the watermarking key, although such adversaries are defined as fair in the framework of [6]. The authors argue that the leaked information will make it easier for an unfair adversary to attack the system's robustness (degrade the watermark channel) and, therefore, use it as a measure for the security of watermarking schemes. This intuition is likely to hold in most cases, but it is important to note that the converse does not hold, i.e., there are watermarking schemes with poor robustness, but which do not leak

[3] The mutual information $I(X;Y)$ between X and Y is defined as the reduction of entropy that Y provides about X.

[4] Kalker models a watermarking scheme as a multiplexed communication system that multiplexes the original data channel and the watermark channel.

[5] Therefore, security, according to Kalker's definition, implies robustness.

any information on the secret watermarking key.[6] Hence, we conclude that the information leaked on the secret watermarking key is not a suitable measure for the robustness/security of the watermarking scheme. Furthermore, one cannot formally define and distinguish between fair and unfair adversaries.

Cayre et al. [7] focus on the security of watermarking schemes and do not consider security against application-level attacks, such as invertibility and copy-attacks. Although it is a good approach to narrow the definition to cover the essential, distinguishing properties of watermarking schemes only, the definition and measure of security chosen by the authors is too general: they measure the level of security of watermarking scheme in the number of observations (watermarked data) an adversary needs in order to estimate the secret watermarking key. Information leakage is measured using methods from information theory, such as Shannon's mutual information. More concretely, defining the adversary goal is a direct translation of Shannon's definition of security of encryption schemes. According to Cayre et al. [7] "the watermarking technique is perfectly secure if and only if no information about the secret key leaks from the observations". Intuitively, this informal definition seems to be reasonable, but not straightforward to define formally, such that it can be satisfied at all: assume the adversary has observed a triple (W, WM, W'), where the stego-data W' results from embedding watermark WM into the cover-data W, using the secret embedding key $K^{\mathtt{emb}}$. Given these observations, the adversary has a reliable test to recognise the correct secret embedding key: the adversary can run through the whole key space and test for every candidate key, whether $W' \stackrel{?}{=} \mathtt{Embed}(W, WM, K^{\mathtt{emb}})$. This test allows the adversary to rule out most watermarking keys and, obviously, this observation leaks information on $K^{\mathtt{emb}}$. The definition in [7] is mainly motivated by the intuition that "if a watermarking scheme does not provide perfect secrecy, then one would like to measure the information leakage about the key." However, defining watermarking security in terms of secrecy and information leakage about the key is not known to be necessary or sufficient for any meaningful security property of the watermarking scheme:[7] obviously it is *not sufficient* for robustness, because it does not rule out unkeyed non-robust watermarking schemes, e.g., a watermarking scheme that embeds the watermark by substituting all LSBs of an image.[8] Furthermore, this definition applies to applications where the same secret embedding key is used to embed several watermarks into different data: watermarking schemes insecure

[6] As an example consider a watermarking scheme, which uses the secret watermarking key as a one-time-pad to encrypt the watermark and embed it in the LSB of pixels identified by the remainder of the watermarking key. Obviously, this scheme does not leak any information about the watermarking key and the watermark, but can be easily removed by setting the LSB of any pixel to 0.

[7] It holds if there is an arguable equivalence between security and secrecy of the key, which holds for encryption schemes as considered by Shannon.

[8] Even the identity map would fulfil the perfect security definition, as it does not depend on a secret key (private communication with Nicholas Hopper, David Molnar and David Wagner).

according to this definition may nevertheless be secure in other applications, which use a fresh embedding key for every watermark.

Comesaña et al [8,9] closely follow the inadequate notion of security (information leakage) introduced in [6,7] as mentioned above. Their main achievement, compared to [7] is the definition of a new measure to quantify the leaked information and its application to spread spectrum watermarking.

Katzenbeisser [10] also follows this notion of security, suffering from the same problem, but proposes a computational definition of leaked information, which is inspired by computational security definitions for symmetric encryption schemes: the underlying model is a game between the adversary and the honest party (embedding oracle) where the adversary's goal of obtaining information about the key (winning the game) is modeled by his ability to distinguish whether given stego-data was more likely watermarked with one out of two keys, where the cover-data is chosen by the adversary and the actual embedding key is randomly chosen by the embedding oracle.

2.1 Summary and Discussion

We observe that in particular the definition of watermark security remains rather unclear. The main reason is that most researchers tried to define "watermark security" such that it captures any property that may be required by any conceivable application. As applications of watermarking schemes are manifold, posing different requirements on watermarking schemes, it is hard to come up with general definitions and even harder to come up with schemes that fulfil them.

Furthermore, it is more reasonable not to define one *low-granular* term, "watermark security", to comprise different *high-granular* requirements of different applications. High-granular requirements may be secrecy, integrity or authenticity of the watermark, dependency of the watermark on the cover-data (to prevent copy attacks), as well as robustness and collusion tolerance to name only the most important ones. Barely any application (if any at all) requires a single watermarking scheme to provide *all* these high-granular properties[9], beside the fact that it is a difficult, and unnecessary, task to design such watermarking scheme.

Moreover, *high-granular* properties required by certain applications can be attained using cryptographic building blocks on top of the watermarking scheme (layered approach): The secrecy[10] and authenticity/integrity of the watermark can be achieved by applying encryption respectively message authentication codes or digital signatures to the watermark before embedding it. Binding the watermark to the cover-data can be achieved by augmenting the watermark through appending a (robust/perceptual) hash of the cover data to the watermark and authenticating this augmented watermark.

[9] This can be compared to cryptographic hash functions. Some applications require hash functions to be collision-free, while some only require a hash-function to be one-way.

[10] This secrecy property should not be confused with the "steganographic hiding" property, which requires that not even the presence of the watermark can be detected.

We argue that a formal definition of a pure digital watermarking scheme should focus on its distinguishing features, which cannot be achieved by existing, well founded cryptographic primitives. As we argue in later sections these features are the capability of embedding additional information in data, the robustness property as well as detection/extraction errors. Nevertheless, for certain applications it makes sense to require that the watermarking scheme provides further security properties, which, as mentioned above, may be achieved by cryptographic means on the top of the watermarking scheme.

3 Basic Notations and Definitions

Computation Model We write algorithms $O \leftarrow \mathtt{Alg}(I)$ to denote running $\mathtt{Alg}$ on inputs I and assigning the output to variable O. Optional inputs/outputs are set in squared brackets, i.e., in $\mathtt{Alg}(I_1, [I_2])$ the input of I_2 is optional. When we use the term *efficient* in the context of algorithms or computation we mean a Turing Machine with polynomial-time complexity.

Probabilities and Negligible Functions. We denote a probability function with $\mathbf{Prob}[A :: B]$ where A denotes the quantity for which the probability is computed and B the (joint) random variable that induces the underlying probability space.

For example $\mathbf{Prob}[pred(v_2) = \top :: v_1 \xleftarrow{R} \mathbb{Z}_n; v_2 \leftarrow \mathtt{Alg}(v_1)]$ means the probability that predicate *pred* holds on v_2, where the underlying probability space is induced by the random variable consisting of the random variables v_1, uniformly chosen from $\mathbb{Z}_n$, and v_2 which is the random value output by the algorithm $\mathtt{Alg}$ on input v_1. Furthermore, let v be some arbitrary random variable or ensemble of random variables. Then, $[v]$ denotes the *support*, which is the set of all possible values v, i.e. those with non-zero probability.

A *negligible* function $\epsilon(x)$ is a function where the inverse of any polynomial is asymptotically an upper bound, i.e., $\forall d > 0\ \exists x_0\ \forall x > x_0 : \epsilon(x) < 1/x^d$. We denote this by $\epsilon(x) <_\infty 1/poly(x)$.

4 Formal Definition of Watermarking Schemes

4.1 Similarity

A suitable similarity function/predicate is a key aspect in the definition of watermarking schemes and the robustness property. Often, simple distortion metrics, such as the mean squared error (MSE) are used to define similarity. A suitable similarity measure has to consider the semantics and envisaged usage of data: for data such as software, a computational semantics is most suitable, whereas for data consumed by human beings a measure based on models of the human visual/audio system is most suitable (see e.g., Cox et al. [11]). However, the latter may be defined computational as well [12].

In the following we assume a suitable, polynomial-time computable *similarity function/predicate*, also referred to as *similarity test* $sim(W^\star, \hat{W})$, which given

two data items $W^\star$ and $\hat{W}$ outputs $\top$ iff $W^\star$ can be considered sufficiently similar to (according to the usual, agreed semantics) and has been derived from $\hat{W}$. Note that $sim()$ does not need to be symmetric. We have chosen to encapsulate this crucial aspect in a single, general predicate, because it abstracts from the peculiarities of the data types and helps to come up with very clear definitions, based on which one can design and prove applications. In Section 6 we consider the necessary steps when using our definitions to build concrete watermarking schemes with provable robustness.

4.2 Systematics of Watermarking and Robustness Definitions

One has several degrees of freedom when formally defining watermarking schemes and the robustness property. We identified the following orthogonal parameters, which have to be considered carefully, because variation of these parameters leads to significantly different definitions. These parameters concern the type of watermarking scheme (detecting vs. extracting), error probabilities (false-positive and false-negative), all-quantified quantities [11], and adversary model. The latter distinguishes between computational and unconditional (information theoretical) adversaries, the a-priori knowledge of the adversary, active vs. passive adversaries and access to embedding or detection/extraction oracles. We consider these orthogonal parameters in the sequel. Based on the degrees of freedom caused by the variation of these parameters one can establish an *application independent systematic* of definitions for watermarking schemes, similar to the systematic for DL-based cryptographic assumptions introduced in [13]. We consider this as important future work, fertilising both the study of watermarking schemes in a more structured way and the exact specification of requirements of watermark-based applications. For application design one can choose the appropriate definition and watermarking scheme which best suits this application. The following definitions offer formal abstractions of watermarking schemes, which can be used to design and analyse a variety of protocols and applications. Furthermore, compared to previous definitions, our definitions comprise an explicit computational adversary model, including passive and active adversaries, as well as error probabilities of watermarking schemes. We first define the watermarking schemes and introduce error probabilities and robustness later.

Definition 1 (Detecting Watermarking Scheme). *Let $\mathcal{W}$ be the set of cover- and stego-data, let $\mathcal{WM} \subseteq \{0,1\}^+$ be the set of all possible watermarks, let $\mathcal{K}$ be the set of keys and let $par_{sec}^{\mathtt{wm}}$ denote the security parameter of the watermarking scheme. A* (detecting) watermarking scheme *consists of three polynomial-time algorithms:*

[11] When defining properties (of watermarking schemes) it makes a fundamental difference, which items (cover-data, watermark, keys, stego-data) are all-quantified and which of them are assumed to be (randomly) chosen.

As a rule of thumb the more items are all-quantified, the stronger the resulting definition is.

– Key Generation Algorithm: *On input of the security parameters* $par_{sec}^{\mathtt{wm}}$*, the probabilistic key generation algorithm* $\mathtt{GenKey}^{\mathtt{WM}}(par_{sec}^{\mathtt{wm}})$ *generates the* matching keys $(K^{\mathtt{emb}}, K^{\mathtt{det}})$ *required for watermark embedding and detection.*
– Embedding Algorithm: *On input of the cover-data* W*, the watermark* WM *to be embedded and the embedding key* $K^{\mathtt{emb}}$*, the probabilistic embedding algorithm* $\mathtt{Embed}(W, WM, K^{\mathtt{emb}})$ *outputs the watermarked data (stego-data)* W'*, which is required to be perceptibly similar to the cover data* W*. We refer to this requirement as the* intactness property *or* imperceptibility property *and define it formally as:* $\forall W \in \mathcal{W}, \forall WM \in \mathcal{WM}, \forall (K^{\mathtt{emb}}, K^{\mathtt{det}}) \in [\mathtt{GenKey}^{\mathtt{WM}}()]$*:*

$$W' \leftarrow \mathtt{Embed}(W, WM, K^{\mathtt{emb}}) \Longrightarrow sim(W', W) = \top \tag{1}$$

– Detecting Algorithm: *On input of (possibly modified) stego-data* W''*, the watermark* WM*, the original cover-data* W *(optional input), sometimes also referred to as* reference-data *in this context, and the detection key* $K^{\mathtt{det}}$*, the probabilistic*[12] *detection algorithm* $\mathtt{Detect}(W'', WM, [W], K^{\mathtt{det}})$ *outputs a Boolean value* $ind \in \{\top, \bot\}$*. Here,* $\top$ *indicates the presence and* $\bot$ *the absence of the watermark. The detecting watermarking scheme should fulfil a property, which is commonly referred to as the* effectiveness of the watermarking scheme *and which we define as follows:* $\forall W \in \mathcal{W}, \forall WM \in \mathcal{WM}, \forall (K^{\mathtt{emb}}, K^{\mathtt{det}}) \in [\mathtt{GenKey}^{\mathtt{WM}}()]$:

$$W' \leftarrow \mathtt{Embed}(W, WM, K^{\mathtt{emb}}) \Longrightarrow \mathtt{Detect}(W', WM, [W], K^{\mathtt{det}}) = \top \tag{2}$$

The definition of extracting watermarking schemes is similar and has been omitted due to space limitations.

Remark 1. We refer to a watermarking scheme as being *symmetric* iff $K^{\mathtt{det}} = K^{\mathtt{emb}}$. In this case, we usually denote this key as $K^{\mathtt{wm}}$ and refer to it as the *watermarking key*. *Blind* watermarking schemes do not require the cover-data W as an input to $\mathtt{Detect}()$ or $\mathtt{Extract}()$ respectively. A blind watermarking scheme with $K^{\mathtt{det}} \neq K^{\mathtt{emb}}$ is called *asymmetric*.

Remark 2. Sometimes we require an algorithm that represents the sampling/choice of a watermark $WM \in \mathcal{WM}$ by the application. We denote this sampling by $WM \leftarrow \mathtt{GenWM}(par_{sec}^{\mathtt{wm}})$ and stress that $\mathtt{GenWM}()$ does not generate the watermark *signal*, but rather the encoded watermark message. Therefore, $\mathtt{GenWM}()$ is not part of the watermarking scheme, but rather a part of the application.

4.3 Error Probabilities of Watermarking Schemes

So far we did not allow the watermark detector/extractor to err, which is both a strong requirement and unrealistic in practice: As most practical watermarking schemes rely on statistical tests, their outputs inherently involve uncertainties and may be incorrect with a certain probability. Furthermore, for most applications a

[12] Although the majority of detection algorithms is not probabilistic we model detection as an probabilistic algorithm to make our definition as general as possible.

negligible error probability may be tolerated. For detecting watermarking schemes we distinguish two types of errors: *false-positive errors* and *false-negative errors*. Informally speaking, a false-positive error means that the detection algorithm indicates a watermark to be present, although it has actually not been embedded, whereas a false-negative error means that the detection algorithm indicates a watermark not to be present, although it actually has been embedded.

When using watermarking schemes as building blocks in protocols or other applications, these errors occur with certain probabilities, which result from the probability distribution of the watermark detector/extractor's inputs in that specific application environment. As these error probabilities are crucial to the performance of the overall applications or protocols, we will discuss them in more details and formalise them in the sequel. The formalisation will be exemplarily done for detecting watermarking schemes and we note that the definitions for extracting watermarking schemes are analogous.

Definition 2 (False-Positives). *We call an input tuple* $(W'', WM, W, K^{\mathtt{wm}})$ *to the detection algorithm a* positive *iff* $\mathtt{Detect}(W'', WM, W, K^{\mathtt{wm}}) = \top$. *A* false-positive *is a tuple* $(W'', WM, W, K^{\mathtt{wm}})$ *with:*

$$\mathtt{Detect}(W'', WM, W, K^{\mathtt{wm}}) = \top$$
$$\wedge \nexists W' : (W' \in [\mathtt{Embed}(W, WM, K^{\mathtt{wm}})] \wedge W'' \in \{\hat{W} | sim(\hat{W}, W')\})$$

We define the *positives set* of a watermarking scheme as the set of all input tuples $(W'', WM, W, K^{\mathtt{wm}})$ yielding a positive detection result (*positive tuple*) $\mathcal{PS} := \{(W'', WM, W, K^{\mathtt{wm}}) \mid \mathtt{Detect}(W'', WM, W, K^{\mathtt{wm}}) = \top\}$ and we define $\mathcal{FPS}$ as the set of all false-positives. Furthermore, we define the *positives rate* as the fraction of positive tuples to all such tuples $pr := |\mathcal{PS}|/|\mathcal{W} \times \mathcal{WM} \times \mathcal{W} \times \mathcal{K}|$ and, similarly, $fpr := |\mathcal{FPS}|/|\mathcal{W} \times \mathcal{WM} \times \mathcal{W} \times \mathcal{K}|$. Note that these rates are completely determined by the watermarking scheme and does not depend on the application context, in which the watermarking scheme is being used. In contrast, the *positives probability* and *false-positive probability* are not completely determined by the watermarking scheme, but additionally depend on the probability distribution of works, watermarks and watermarking keys (see [11]), *which itself depends on the context given by the application in which the watermarking scheme is being used.* In particular, the application's security requirements (or conversely the adversary's goals) and the underlying trust model play a central role in defining an adequate positives probability. Depending on the above aspects, one can distinguish several different types of positives probabilities. Here, we focus on *adversarial positives probabilities*: in most security applications, at least parts of the input tuple to `Detect`() can be computed freely by the adversary (without adhering to a pre-defined distribution), such that it triggers the detector and, in addition, fulfils a certain application dependent predicate `side_condition`.[13] We refer to these positive probabilities as *adversarial*

[13] In case of dispute resolving applications, the side-condition predicate may state that the false-positive watermark, computed by the adversary, is also detectable in the original work of the rightful author, thus leading to a deadlock.

false-positives probabilities and distinguish several adversarial false-positives probabilities, which vary depending on the a-priori information available to the adversary and the side-conditions the positives have to fulfil. Both strongly depend on the concrete application scenario, denoted as `application`, in which the watermarking scheme is used.

Definition 3 (General Adversarial False-Positive Probability). *Let $\mathcal{A}$ denote the adversary algorithm. We define the* general adversarial positives probability $pp_{adv}(\mathcal{A})$ *as follows:*

$$\begin{aligned}&\textbf{\textit{Prob}}[(\texttt{Detect}(W', WM_{\mathcal{A}}, W_{\mathcal{A}}, K^{\texttt{wm}}_{\mathcal{A}}) = \top) \wedge \\ &\texttt{side\_condition}((W, WM, K^{\texttt{wm}}, W'), (W_{\mathcal{A}}, WM_{\mathcal{A}}, K^{\texttt{wm}}_{\mathcal{A}}, W'_{\mathcal{A}})) :: \\ &\quad (W, WM, K^{\texttt{wm}}, W') \leftarrow \texttt{application}; \\ &\quad (WM_{\mathcal{A}}, W_{\mathcal{A}}, K^{\texttt{wm}}_{\mathcal{A}}, W'_{\mathcal{A}}) \leftarrow \mathcal{A}([W], [WM], [K^{\texttt{wm}}], [W'], par^{\texttt{wm}}_{sec});]\end{aligned}$$

Note that, in contrast to the non-adversarial positives probabilities, the adversarial positives probabilities depend on the concrete adversary strategy (formalised by the algorithm $\mathcal{A}$), which the adversary employs to compute the positive tuple. Furthermore, one has several degrees of freedom regarding the a-priori information given to the adversary. We modelled this by defining the inputs to $\mathcal{A}$ as optional parameters. The adversarial false-positive probability has often been neglected in the design of security critical applications, such as dispute-resolving schemes and further copyright protection protocols, mostly because its impact on the security of the overall protocol has been underestimated.[14] It is obvious that in any application where the presence of watermarks serves as evidence, such as dispute resolving, authorship proofs or fingerprinting the false-positive probability becomes critical: the higher the false-positive probability is, the lower is the "conclusiveness" or "reliability" of a detected/extracted watermark as a piece of evidence. Finally, we want to note that it is difficult to actually determine these adversarial error probabilities or bound them from above for concrete watermarking schemes. Therefore, assumptions about upper bounds of these probabilities are very strong assumptions.

Definition 4 (Negatives Rate). *We define a* negative *as a tuple* $(W'', WM, W, K^{\texttt{wm}})$*, yielding a negative detection result, i.e.,* $\texttt{Detect}(W'', WM, W, K^{\texttt{wm}}) = \perp$*. Furthermore, we define the* negatives set *of a watermarking scheme as the set of negative tuples* $\mathcal{NS} := \{(W'', WM, W, K^{\texttt{wm}}) \mid \texttt{Detect}(W'', WM, W, K^{\texttt{wm}}) = \perp\}$ *and we define the* negatives rate *as the fraction of negative tuples to all such tuples:* $nr := |\mathcal{NS}|/|\mathcal{W} \times \mathcal{WM} \times \mathcal{W} \times \mathcal{K}| = 1 - pr$.

More interesting is the definition of *false*-negatives, for which we require an appropriate notion of when an input tuple should actually be a positive tuple,

[14] In [11] (p. 30), Cox et al. state: "In the case of proof of ownership, the detector is used so rarely, that a probability of 10^{-6} should suffice to make false positives unheard of." Here, "probability" refers to random non-adversarial probability, which makes it quite easy for an adversary to compute false-positives, fulfilling certain side-conditions and, thereby, breaking the security of (see [14] for example.).

which itself *depends on the properties required from the watermarking scheme*: robust watermarking schemes require that the inputs of a run of the embedding algorithm $(W, WM, K^{\mathtt{wm}})$ plus the stego-data, resulting from this run, *or any similar data, derived from the stego data*, is a positive. Following this view, a false-negative is always a breach of robustness.

Definition 5 (False-Negatives and False-Negative Rate). *For a robust watermarking scheme a* false-negative tuple *is a tuple* $(W'', WM, W, K^{\mathtt{wm}})$ *with* $\mathtt{Detect}(W'', WM, W, K^{\mathtt{wm}}) = \perp$*, although* W'' *has been derived from watermarked data* $W' \leftarrow \mathtt{Embed}(W, WM, K^{\mathtt{wm}})$ *and* $sim(W'', W') = \top$ *holds, i.e., detection should be successful by the robustness property of the watermarking scheme.*

Let $\mathcal{FNS}$ *denote the set of all false-negatives. We define the false-negative rate as the fraction of false-negatives and the set of tuples that should trigger a perfectly robust detector according to our robustness definition:* $fnr := |\mathcal{FNS}|/|\mathcal{W} \times \mathcal{WM} \times \mathcal{W} \times \mathcal{K}|$.

Analogous to the discussion above, one can define *adversarial false-negative probabilities*, denoting the probability that an adversary can compute a false-negative tuple. Due to lack of space and its relation to the robustness definition we omit this definition here. In the following Section we formalise *"robust watermarking schemes"*, which can be seen as an extension of the effectiveness property to those data, which has been derived from the stego-data and is still sufficiently similar to it.

5 Computational Robustness Definitions

5.1 Robustness Against Passive Adversaries

Informally, the robustness property against passive adversaries states that a watermark should remain detectable, even if the stego-data has been (maliciously) modified. Clearly, detectability (or extractability) cannot be guaranteed for any modification[15]. Therefore, the correct informal characterisation of a *robust* watermarking scheme is that it can detect/extract the watermark even in a (maliciously) modified stego-data *as long as the stego-data is perceptibly similar to the cover-data.*

The robustness property is of great importance, especially in the context of copyright protection applications, because the detectability of embedded watermarks is crucial for the overall system security. Unfortunately, robustness is not well understood so far. Most researchers give informal characterisations of robustness or define it as resistance against an inherently incomplete list of known attacks [15,16,17]. Cox et al. [11] distinguish between *robustness* and *security* of watermarking schemes: they characterise "robustness" as the "the ability to detect the watermark after common signal processing operations", whereas they

[15] Consider for example a modification, which completely garbles the stego-data or transforms it to the constant bit-string 1^n.

refer to "security" as the "ability to resist hostile attacks". As we address watermarks exclusively in the context of security critical applications, this distinction would be artificial and we define robustness to cover also the ability to resist hostile removal-attacks.

Definition 6 (Symmetric Computational Robustness). *We define a symmetric watermarking scheme to be* computationally robust*, iff*

$$
\begin{aligned}
&\forall WM \in \mathcal{WM}, \forall \textit{ probabilistic polynomial-time adversary } \mathcal{A} \\
&\boldsymbol{Prob}[\mathtt{Detect}(W'', WM, W, K^{\mathtt{wm}}) = \perp \quad \wedge \quad sim(W'', W') = \top :: \\
&\quad W \leftarrow \mathcal{W}; \\
&\quad K^{\mathtt{wm}} \leftarrow \mathtt{GenKey}^{\mathtt{WM}}(par_{sec}^{\mathtt{wm}}); \\
&\quad W' \leftarrow \mathtt{Embed}(W, WM, K^{\mathtt{wm}}); \\
&\quad W'' \leftarrow \mathcal{A}(W', [WM], par_{sec}^{\mathtt{wm}});] \\
&< _{\infty}\ 1/poly(par_{sec}^{\mathtt{wm}})
\end{aligned}
$$

Informally, this means that symmetric watermarking scheme is called robust, iff it is computationally infeasible for an adversary, given watermarked data W' and the watermark WM, to produce perceptibly similar data W'', in which the same watermark WM cannot be detected anymore.

When designing an application, one has to choose the correct robustness definition. Especially the input available to the adversary $\mathcal{A}$ depends on the context of the target application: In applications such as dispute resolving, it is reasonable to assume that the adversary does not know the watermark. However, in applications such as copy protection, there exists only a small set of possible watermarks (e.g., "copy permitted", "do not copy") and therefore, it is more realistic to assume that $\mathcal{A}$ gets WM as an additional input. In general, the more inputs the robustness definition allows the adversary to use, the stronger it is (and the more difficult it is for a watermarking scheme to fulfil it). As a consequence, the following robustness definition for asymmetric watermarking schemes is even harder to achieve than that for symmetric schemes, because the adversary is granted access to the watermark and detection key as well[16].

Definition 7 (Asymmetric Computational Robustness). *An* asymmetric *watermarking scheme is called robust, iff*

$$
\begin{aligned}
&\forall WM \in \mathcal{WM}, \forall \textit{ probabilistic polynomial-time attacker } \mathcal{A} \\
&\boldsymbol{Prob}[\mathtt{Detect}(W'', WM, W, K^{\mathtt{det}}) = \perp \quad \wedge \quad sim(W'', W') = \top :: \\
&\quad W \leftarrow \mathcal{W}; \\
&\quad (K^{\mathtt{emb}}, K^{\mathtt{det}}) \leftarrow \mathtt{GenKey}^{\mathtt{WM}}(par_{sec}^{\mathtt{wm}}); \\
&\quad W' \leftarrow \mathtt{Embed}(W, WM, K^{\mathtt{emb}}); \\
&\quad W'' \leftarrow \mathcal{A}(W', WM, K^{\mathtt{det}}, par_{sec}^{\mathtt{wm}});] \\
&< _{\infty}\ 1/poly(par_{sec}^{\mathtt{wm}})
\end{aligned}
$$

The robustness definition for asymmetric schemes is very similar to that of symmetric schemes. The main difference is that the adversary additionally receives

[16] Amongst others, this provides the adversary with a detection oracle.

the public detection inputs (detection key and watermark). Alternatively, one may define robustness of asymmetric watermarking schemes by providing the adversary only with W' and the public detection key. However, from the application's perspective, it does not make sense to make the detection key publicly available, without at the same time making the watermark publicly available. Therefore, we have chosen to provide the adversary with the watermark as well. Amongst others, this definition is suitable for copy control applications.

5.2 Robustness Against Active Adversaries

Early definitions of robustness and watermark security did not consider active adversaries, interacting with and, thereby, having indirect access to the embedder and detector, *including* the corresponding keys. As a matter of fact, these robustness definitions may be too weak for many applications of watermarking schemes.[17] Therefore, it is crucial to also consider robustness against active adversaries and have suitable definitions on-hand. Hence we desire to model adversaries that have access to the functionality of some public algorithm, initialised with some secret system parameter (e.g., the secret embedding or detection key), but without having direct access to this secret parameter. The common technique to model them is to provide adversaries access to oracle machines. The secret system parameter, used to initialise the oracle, is usually generated by the correct party according to the rules of the two party game underlying the respective computational security definition. Oracle machines can be restricted to answer a limited number of t, polynomially bounded in the security parameter, queries only. Such oracles are referred to as t-oracles. Actually, this "free" access to oracles is more than one would expect in most application settings, because there, the honest party, indirectly granting access to the embedder or detector, would usually not blindly apply it to any input data without some predefined verifications. [18] However, by modelling active attacks by granting *free* access to oracles, the definition becomes application independent and one is *on the safe side*, because this guarantees that one can design applications without implementing further checks to limit access to the oracle (e.g., copy-protection in CE devices).

In analogy to the core algorithms of a watermarking scheme, i.e., the embedding and detection/extraction algorithm, we define two types of oracles, embedding oracles and detection/extraction oracles. Furthermore, one can distinguish

[17] Consider a dispute resolving scheme as an example (see [18] for an overview): here the author has to prove the presence of the watermark in the disputed work to a judge. As the disputed work usually has been generated by the adversary, the adversary has indirect, restricted access to the watermark detector, which, obviously, has to be modelled in the robustness definition.

[18] Consider watermark-based copy control as an example: the licensing authority might perform certain tests to make sure to embed a "copy freely" watermark only in reasonable looking data, such as to not compromise the security of the watermarking scheme, whereas the detector, embedded in a low-cost DVD recorder provides access to an unlimited detection oracle.

several kinds of embedding oracles according to the *secret information contained in the oracles* and *the form of queries answered by this oracle*. The most usual embedding oracles are discussed below:

1. **Embedding oracles with secret embedding key:** These embedding oracles are initialised with a secret watermark embedding key $K^{\mathtt{emb}}$, as provided by the application/correct party in the security definition and answers t queries of the form $(W_{\mathcal{A}}, WM_{\mathcal{A}})$. We denote such oracles, initialised with $K^{\mathtt{emb}}$, as $\mathcal{O}^t_{\mathtt{Embed},K^{\mathtt{emb}}}$. Given a query $(W_{\mathcal{A}}, WM_{\mathcal{A}})$ oracle $\mathcal{O}^t_{\mathtt{Embed},K^{\mathtt{emb}}}$ replies with answer $W'_{\mathcal{A}} \leftarrow \mathtt{Embed}(W_{\mathcal{A}}, WM_{\mathcal{A}}, K^{\mathtt{emb}})$.
2. **Embedding oracles with secret embedding key and secret watermark:** Another type of embedding oracle, considered here, is initialised with a secret embedding key and a secret watermark and answers queries of the form $(W_{\mathcal{A}})$: given a query $(W_{\mathcal{A}})$ the embedding oracle $\mathcal{O}^t_{\mathtt{Embed},K^{\mathtt{emb}},WM}$ replies with answer $W'_{\mathcal{A}} \leftarrow \mathtt{Embed}(W_{\mathcal{A}}, WM, K^{\mathtt{emb}})$.

Similarly, one can define several types of detection/extraction oracles, depending on the secret oracle initialisation information and the form of queries answered by the detection oracle: the first type of detection oracle $\mathcal{O}^t_{\mathtt{Detect},K^{\mathtt{det}}}$ is initialised with a secret detection key $K^{\mathtt{det}}$ and answers queries of the form $(W'_{\mathcal{A}}, WM_{\mathcal{A}})$, whereas the second type $\mathcal{O}^t_{\mathtt{Detect},K^{\mathtt{det}},WM}$ is initialised with a fixed detection key $K^{\mathtt{det}}$ *and a fixed secret watermark* WM and answers queries of the form $(W'_{\mathcal{A}})$. In asymmetric watermarking schemes, as defined above, the adversary is supposed to know the public detection key, which provides him with "unlimited access to an detection oracle". Therefore, to model active attacks against asymmetric watermarking schemes, one only has to consider embedding oracles. We denote an adversary, having oracle access to a set of oracles $\mathcal{O}_1, \ldots, \mathcal{O}_n$ as $\mathcal{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_n}$. Finally, we define a symmetric watermarking scheme to be *computationally robust against active adversaries with access to an embedding and detection oracle*, iff

$\forall WM \in \mathcal{WM}, \forall$ prob. polynomial-time attacker $\mathcal{A}$
$\mathbf{Prob}[\mathtt{Detect}(W'', WM, W, K^{\mathtt{wm}}) = \bot \quad \wedge \quad sim(W'', W') = \top ::$
$W \leftarrow \mathcal{W}$;
$K^{\mathtt{wm}} \leftarrow \mathtt{GenKey}^{\mathtt{WM}}(par^{\mathtt{wm}}_{sec})$;
$W' \leftarrow \mathtt{Embed}(W, WM, K^{\mathtt{wm}})$;
$W'' \leftarrow \mathcal{A}^{\mathcal{O}^t_{\mathtt{Embed},K^{\mathtt{emb}},WM}, \mathcal{O}^t_{\mathtt{Detect},K^{\mathtt{det}},WM}}(W', par^{\mathtt{wm}}_{sec});]$
$<_\infty 1/poly(par^{\mathtt{wm}}_{sec})$

6 Conclusion and Cautionary Note

Formal definitions for security properties of watermarking schemes are crucial when proving the security of multimedia applications that combine cryptographic methods with watermarking. Existing literature on formal security definitions for watermarking is not extensive and still has conceptual shortcomings. In this paper, we discussed these shortcomings as well as the subtle

aspects/parameters that existing proposals do not cover. We proposed a formal framework and definitions for watermarking schemes that incorporate these aspects/parameters and can be used as a suitable abstraction for security proofs of multimedia security schemes.

Finally, we stress that currently no watermarking scheme is known to fulfil the computational robustness definitions as defined above. Thus, the robustness assumption is a stronger assumption compared to standard number-theoretical assumptions in cryptography: Number-theoretical assumptions have shown their reasonability, since no efficient algorithms solving them have been found for a long period of time. In contrast, any watermarking scheme proposed so far and claimed to be robust fails to fulfil the computational robustness definition. This leaves us with a gap between our abstract model of robust watermarking schemes and the watermarking schemes available today. Nevertheless our formal definitions provide an appropriate abstraction (similar to the *marking assumption* in fingerprinting [19]) which can be used to design secure applications, such as dispute-resolving protocols.[19]

On the other hand, based on our work, provably robust watermarking schemes may be developed as follows: First, we have to define $sim()$ for the respective data type, which, based on the current understanding of the HVS, is a hard task for multimedia data. However, for data such as software, having a well-defined formal semantics, it seems to be feasible to come up with a suitable definition. Second, we have to choose a suitable computationally hard problem on the respective data type. For software, such problems are well-known for a long time [20] and also considered in the design of software obfuscation. Third, we have to define the watermarking scheme, such that embedding preserves similarity and such that an attacker, being able to break the scheme's robustness, can be used to efficiently and accurately solve a hard problem (proof by reduction).

References

1. Cachin, C.: An information-theoretic model for steganography. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 306–318. Springer, Heidelberg (1998)
2. Mittelholzer, T.: An information-theoretic approach to steganography and watermarking. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 1–16. Springer, Heidelberg (2000)
3. Hopper, N.J., Langford, J., von Ahn, L.: Provably secure steganography. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, Springer, Heidelberg (2002)
4. Backes, M., Cachin, C.: Public-key steganography with active attacks. Report 2003/231, Cryptology ePrint Archive (2003)
5. Kalker, T.: Considerations on watermark security. In: IEEE International Workshop on Multimedia Signal Processing (MMSP'01) 2001, pp. 201–206 (2001)
6. Barni, M., Bartolini, F., Furon, T.: A general framework for robust watermarking security. Signal Processing, pp. 2069–2084 (2003)

[19] In fact, the marking assumption is even stronger than the robustness assumption, as it requires robustness against adversaries in possession of differently marked versions of the same cover-data.

7. Cayre, F., Fontaine, C., Furon, T.: Watermarking security, part one: theory. In: IS&T/SPIE International Symposium on Electronic Imaging 2005. In: Proceedings of the SPIE., SPIE (2005) pp.746–757. Security, Steganography, and Watermarking of Multimedia Contents VII (2005)
8. Comesaña, P., Pérez-Freire, L., Pérez-González, F.: An information-theoretic framework for assessing security in practical watermarking and data hiding scenarios. [21]
9. Comesaña, P., Pérez-Freire, L., Pérez-González, F.: Fundamentals of data-hiding security and their application to spread spectrum analysis. In: Barni, M. (ed.) IH 2005. LNCS, vol. 3727, pp. 146–160. Springer, Heidelberg (2005)
10. Katzenbeisser, S.: Computational security models for digital watermarks. [21]
11. Cox, I., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann Publisher, San Francisco (2002)
12. Tran, N.: Hiding functions and computational security of image watermarking systems. In: 15th IEEE Computer Security Foundations Workshop, pp. 295–306. IEEE Computer Society Press, Los Alamitos (2002)
13. Sadeghi, A.R., Steiner, M.: Assumptions related to discrete logarithms: Why subtleties make a real difference. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 243–260. Springer, Heidelberg (2001)
14. Adelsbach, A., Katzenbeisser, S., Sadeghi, A.R.: On the insecurity of non-invertible watermarking schemes for dispute resolving. In: Kalker, T., Cox, I., Ro, Y.M. (eds.) IWDW 2003. LNCS, vol. 2939, pp. 355–369. Springer, Heidelberg (2004)
15. Swanson, M.D., Kobayashi, M., Tewfik, A.H.: Multimedia data-embedding and watermarking technologies. In: Proceedings of the IEEE, vol. 86 (1998)
16. Katzenbeisser, S., Petitcolas, F.A.: Information Hiding: techniques for steganography and digital watermarking. Artech House Publishers (2000)
17. Hartung, F., Kutter, M.: Multimedia watermarking techniques. In: Proceedings of the IEEE, Special Issue on Identification and Protection of Multimedia Information, vol. 87, pp. 1079–1107 (1999)
18. Adelsbach, A., Sadeghi, A.R.: Advanced techniques for dispute resolving and authorship proofs on digital works. In: Proceedings of SPIE, Security and Watermarking of Multimedia Contents V, vol. 5020 (2003)
19. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 452–465. Springer, Heidelberg (1995)
20. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M.: Complexity and Approximation. Springer-Verlag, Berlin Germany (1999)
21. Piva, A. (ed.): In: 6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005), Special Session on Media Security (2005)