A general framework for robust watermarking security

Mauro Barni, Franco Bartolini, Teddy Furon

A general framework for robust watermarking security

Mauro Barni [a], Franco Bartolini [b,2], and Teddy Furon [c]

[a] *Department of Information Engineering, University of Siena*
Via Roma, 56 - 53100 Siena, Italy
e-mail: barni@dii.unisi.it

[b] *Department of Electronics and Telecommunications, University of Florence*
Via di S. Marta, 3 - 50139 Firenze, Italy
e-mail: barto@lci.det.unifi.it

[c] *TEMICS project, IRISA/INRIA*
Campus de Beaulieu, 35000 Rennes, France
e-mail: teddy.furon@irisa.fr

Abstract

The analysis of the security of watermarking algorithms has received increasing attention since it has been recognized that the sole investigation of robustness issues is not enough to properly address the challenges set by practical applications. Such a security analysis, though, is still in its infancy, up to a point that a general agreement has not yet been reached even on the most fundamental problems. The purpose of this paper is to provide a general security framework encompassing most of the problems encountered in real-world applications. By considering the amount of information the attacker has about the watermarking algorithm, we introduce the notion of fair and un-fair attacks, so to ease the classification of different systems and attacks. Though we recognize that many important differences exist between watermarking and cryptographic security, a large part of our work is inspired by the Diffie-Helmann's paradigm, which is widely used in cryptography. For each class of systems great care is taken to describe both the attacker's and watermarker's point of view, presenting the challenges raised by each system to these different actors. Finally, we try to outline some research directions which, according to us, deserve further analysis.

Key words: Digital watermarking, watermarking attacks, watermarking security, fair and unfair attacks, Diffie-Hellmann's attacks classification.

[1] Authors' names are listed in alphabetical order.
[2] Contact author: Franco Bartolini.

1 Introduction

Although most of watermarking research has focused on robustness, capacity, and perceptibility issues, it has been recently acknowledged that security aspects are also (if not even more) important for many secure applications such as copy control [1,2], ownership verification [**?**], and authentication [**?**]. Recognizing that the development of secure robust watermarking schemes, and even that the exact definition of what security means in a watermarking context, is still in its infancy, it is the objective of this paper to survey the most important problems that have been raised, as well as the solutions that have been proposed so far with regard to this topic in the literature.

1.1 Watermarking is an application driven solution

The importance of the security aspects of a watermarking technique is highly related to the application the technique has been devised to serve. There are applications for which security does not constitute a problem (e.g. document labelling, content enhancement), and among the applications for which security is an issue, different levels of safety can be identified (*e.g.* in DRM [3] mechanisms, while the CPTWG [4] looked for a watermarking technique that, by making hacking slightly difficult, would only help 'keep honest people honest', at the same time, the SDMI [5] was aiming at a hacker-proof watermarking solution [4]).

Another caveat is the fact that each application uses watermarking for a particular purpose and in a specific framework. A common mistake is a misunderstanding of the functionality offered by a watermarking technique. Watermarking is widely but wrongly believed to be the art of hiding owners' name in their contents. This is not true. The scope of potential applications is broader than copyright protection and proof of ownership. On the other hand, focusing on this latter application, watermarking may not be the solution. The owner may not be the only one to embed data within his works; usurpers also build their own private channel. As it is, watermarking doesn't provide the owner a solution to copyright struggles [5]. This lack of understanding stems from the fact that, from a security point of view, watermarking is, at best, just a security brick. Computer Security people usually name this a *primitive*. This element is useless on its own unless included in a global system: that is a structured set of primitives providing a solution to a certain problem. It turns out that security analysis is then, above all, application driven. There is also

[3] Digital Rights Management
[4] Copy Protection Technical Working Group [3]
[5] Secure Digital Music Initiative

a wide range of working frameworks. For instance, it is different the case of copy control mechanisms, where the detector has to be considered to as public [1,2] (*i.e.* a system embedded in consumer electronic devices), and ownership verification systems, where the detector is private [6](*i.e.* used by a trusted person).

Due to its versatility of use in functionality and framework, it seems that each watermarking application should require a dedicated security analysis. In this paper, we make an effort to decouple the application impacts presenting a methodology to generally tackle security analysis. Hence, we do not refer to any particular application, but when relevant, anyway, related applications illustrate the considered situation.

1.2 Definition of security of robust watermarking

Security of watermarking based applications can be obviously faced at a protocol level, for example by integrating watermarking systems with cryptographic techniques [7]: we do not deal with this kind of solutions, because, although very important and effective, they do not regard watermarking technology only. Our attention is thus solely concentrated on the signal processing aspects of watermarking security.

In particular, we focus on robust watermarking. In this context, a watermarking algorithm aims at mixing a non-perceptible communication channel with multimedia data, in such a way that the capacity of this extra channel degrades smoothly with the distortion the watermarked content undergoes [8]. The smoother the capacity function versus the distortion due to content manipulations, the more robust the watermarking technique. Then, robust watermark security refers to the inability by unauthorized users to access the extra channel. It ensures adversaries can not emit or decode hidden bits, and destroy this channel. As we can assume that the first two threats are satisfactorily addressed by cryptographic primitives, the main concern is the fact that, by exploiting the knowledge of the particular system or keys used to watermark the content [6], the adversary can degrade the channel capacity much more efficiently than robustness analysis could let imagine [8,9].

[6] Such an information may be publicly available at the attacker, or may be acquired through particular attacks aimed at getting such a, supposedly secret, information.

Although watermarking pertains to signal processing, it is also related to computer security, whence, the comparison with cryptography must be properly treated. There are many differences between cryptography and robust watermarking techniques, beginning from the very objective of each technology. While in the first case, as far as encryption is concerned, the goal is to make the semantic of a communication not understandable from a possible opponent assuming that no deterioration of the message carrier happens, in the latter case the aim is to protect the hidden communication itself from possible deterioration of the channel (*i.e.* of the cover data). Anyway, an investigation on how evaluation of cryptographic algorithms is proceeded is compelling [10]. We borrow that approach for understanding the security issues of the robust watermarking tool. In particular, as it is done for cryptography since Diffie and Hellmann's article [11], the security analysis is driven by the data the opponent observes once the embedder starts producing watermarked contents.

Only watermarked content. The attacker can only have access to one or more watermarked documents.

Chosen watermarked content. The attacker can choose one or more (pretended) watermarked documents.

Original and watermarked pair. The attacker can have access to one or more pairs of original and corresponding watermarked documents.

Chosen original and watermarked pair. The attacker can choose one or more pairs of original and corresponding watermarked documents.

The first attack is the most important as every watermarking system has obviously to deal with it. The second one is mainly related to the possibility for the attacker to have access to the decoding process. He observes the outputs of the detector for some selected documents. The third case reflects the possibility to have original documents available to the adversary. The fourth attack can be implemented if the watermark embedding system is available to the pirate, so that he can generate original and watermarked pairs. In the second and fourth cases, it is assumed that the pirate has not access to the embedding or decoding key: either the device is left unarmed without the secret key, or this key is hard wired in the device considered then as a black sealed box.

The main advantage of this classification is that it decouples analysis from the application. Theoretical research on watermarking proves evaluation of the security level of a technique for each class of attacks. There are techniques more secure than others for a given class of attacks, but weaker for another class. Once an application is targeted, a practical watermarking designer analyzes which type of attack is a real threat in this particular framework. There exist indeed very few applications where the four classes of observations are available

to the pirate. Then, the designer selects the technique, which is the most robust and secure to these potential threats.

1.4 Notations

For our purposes, the following notations are introduced. We denote an original content, its watermarked version and its watermarked and attacked version by c_o, c_w, and c_a. The embedding function Emb(.) receives four arguments: the algorithm a, the message to be hidden m, the embedding key k_{E} and the original content c_o, and produces the watermarked content c_w:

$$c_w = \mathrm{Emb}(c_o, a, m, k_{\mathrm{E}}) \tag{1}$$

The watermark embedding algorithm is modelled as a three step process. First N features are extracted from the document and stored in a feature vector $\vec{f_o} = \mathrm{Ext}(c_o, a)$. The watermarked feature vector is the mixing of the original feature vector with the watermark signal $\vec{w}$: $\vec{f_w} = \mathrm{Mix}(\vec{f_o}, \vec{w})$. Note that $\vec{w}$ may depend on $\vec{f_o}$ if the informed embedding approach is used [12]. Finally, these modified features are mapped back in the original document: $c_w = \mathrm{Ext}^{-1}(\vec{f_w}, c_o, a)$. The watermark signal has N samples, which are function of the embedding key and the message to be hidden: $\vec{w} = \mathrm{Gen}(m, k_{\mathrm{E}}, a)$.

In the same way, the decoding function Dec(.) yields a message from a received content as follows:

$$\hat{m} = \mathrm{Dec}(c, a, k_{\mathrm{D}}). \tag{2}$$

A distinction is needed between the decoding of a hidden message and the detection of a watermark signal. In the first case, m belongs to a message space $\mathcal{M} = \{1, \ldots, 2^C\}$, where C is the capacity in bits. In the latter case, $m \in \{0, 1\}$ where $m = 0$ ($m = 1$) is not a symbol to be hidden but it reflects the fact that the content has not been watermarked (resp. it has been watermarked).

Finally, $\{A, C_o, M, K_{\mathrm{E}}, K_{\mathrm{D}}\}$ are random variables, whereas $\{a, c_o, m, k_{\mathrm{E}}, k_{\mathrm{D}}\}$ denotes one instanciation of these random variables.

The observations made by the opponent since the embedder started to produce watermarked contents, are denoted by $\mathcal{O}$. More explicitly, for the four classes mentioned previously, we have:

Only watermarked content. $\mathcal{O} = \{c_{w,i}\}_{i \in \mathcal{IO}}$.
Chosen watermarked content. $\mathcal{O} = \{c_{w,i}, \hat{m}_i\}_{i \in \mathcal{IC}}$.
Original and watermarked pair. $\mathcal{O} = \{c_{w,i}, c_{o,i}\}_{i \in \mathcal{IO}}$.
Chosen original and watermarked pair. $\mathcal{O} = \{c_{w,i}, c_{o,i}\}_{i \in \mathcal{IC}}$.

$\mathcal{IO}$ represents a random set of content's indices, whereas $\mathcal{IC}$ is the set of the indices of the contents chosen on purpose by the opponent.

This classification certainly needs some refinements to encompass all the types of attacks. For instance, for the only watermarked content attack, the observations can be $\mathcal{O} = \{c_{w,i} = \mathrm{Emb}(c_{o,i}, a, m, k_{\mathrm{E}})\}$, *i.e.* a set of different contents watermarked with the same algorithm, the same embedding key and the same message. This is quite usual in copy protection application [13] or when a fixed template is added to help synchronization [14]. In the case of a linear embedding process, the average attack might give out an accurate estimation of the watermark signal. But, this type of attack also pertains to the case where $\mathcal{O} = \{c_{w,i} = \mathrm{Emb}(c_o, a, m_i, k_{\mathrm{E}})\}$. This is typical from the tracing application, where a fingerprint is inserted in the distributed copies of a work. Then, the collusion attack is a real threat, yielding an unwatermarked copy of the work [15–17].

1.5 Structure of the paper

In this paper we propose a new framework to understand watermarking security: this is based mainly on modelling the watermark as a game with some rules (consisting of the respect of the established secret parameters), and on classifying the attacks as fair, if they obey the rules (i.e. based solely on what is known), or unfair, if they attempt to break the rules (i.e. if they attempt to discover the parameters that the embedder intended to keep secret). The different watermarking approaches will thus be analyzed on the basis of this approach, the security problems and how they can be faced with will be discussed, and the raising challenges pointed out.

The paper is organized as it follows. In section 2 the concept of fair and unfair attacks is introduced and the general framework we will use to analyze watermarking security introduced. In section 3, the classical security-by-obscurity scheme in which security is achieved by keeping all the details of the watermarking algorithm secret is discussed. In section 4, it is assumed that the watermarking algorithm is disclosed, thus letting security rely on the secrecy of the watermarking embedding and decoding keys. The asymmetric watermarking scenario is analyzed in section 5, where the challenges and opportunities set by public-key watermarking are reviewed. The possibility of developing a system in which the attacker knows all the details of the watermarking algorithm is investigated in section 6. The paper ends with some conclusions and suggestions for future research on the topic in section 7.

2 The framework

In the attempt to shed some light about the possible approaches to design a secure watermarking system, and to classify them in a way which is as consistent as possible, we focus on the *a priori* information the pirate is allowed to resort to. According to the, so to say, normal course of the game, such an information is limited by the rule of the game. For instance, we can assume either that the algorithms used to embed and retrieve the watermark is known to the attacker, or that such an information is not available. Let us now assume that the attacker, however ill-intentioned he may be, is a fair player, and as such he obeys the rules of the game. In this case, he tries to make the watermark unreadable being satisfied with the information the rule of the game assigns to him and the observations $\mathcal{O}$ available once the game started. If he is supposed not to know the watermarking algorithm, he operates blindly, whereas if the game rules allow him to access such an information, he tries to design an attack by explicitly exploiting the weaknesses of the particular algorithm used by the owner. In the following, we refer to this kind of attacks, in which the attacker only exploits publicly available information, as *fair* attacks.

Of course, the scenario depicted above is an unrealistic one. In real applications, attackers are obviously not fair, thus they try to access all the information which may be of any help for their goal. For instance, thanks to the observations, they will try to know how the watermarking system works, or to discover the secret keys used for watermark embedding or decoding, even if the rules of the game assume that this is a secret information. From a security point of view, then, it is essential that the watermarker takes care of keeping the secret information secret. This may be a very difficult task, possibly much more difficult than achieving security against fair attacks. It is, then, the self-interest of the watermarker to minimize the information to be kept secret. Of course, as system secretness decreases, the rules of the game tend to favor the attacker, hence making more and more difficult coping with fair attacks. The necessity that the watermarker carefully considers which kind of information is to be kept secret and which information is made publicly available is summarized in figure 1, where the effort to cope with fair attacks and the effort to ensure the secretness of secret information are qualitatively plotted against the amount of to-be-kept-secret information. As it can be seen, the need for a trade-off between security-by-obscurity (right end of the plot) and open-cards watermarking (left end of the plot), readily comes out.

With these remarks in mind, we classify watermarking security analysis according to the *a priori* information $\mathcal{R}$ which is made publicly available to the attacker. More specifically, we consider four scenarios:

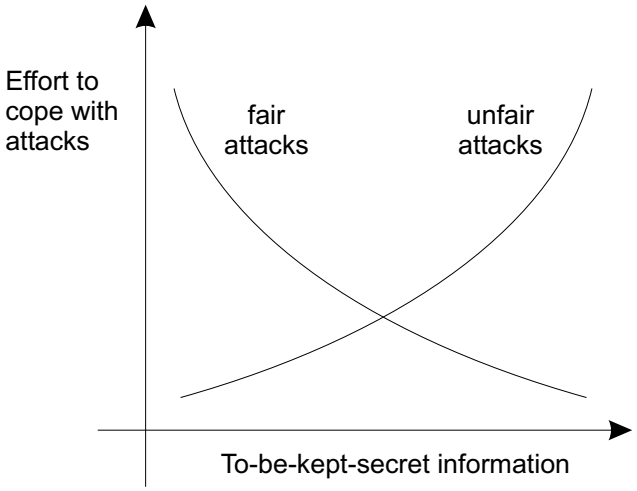

Fig. 1. The security tradeoff. As the amount of to-be-kept-secret information increases, the effort to cope with unfair attacks increases and that to cope with fair attacks diminishes.

No knowledge : $\mathcal{R} = \emptyset$,
Knowledge of the embedding and detection algorithms : $\mathcal{R} = \{a\}$,
Knowledge of the detection key : $\mathcal{R} = \{a, k_{\mathrm{D}}\}$,
Knowledge of the detection and the embedding keys : $\mathcal{R} = \{a, k_{\mathrm{D}}, k_{\mathrm{E}}\}$.

Note that we explicitly make provision for asymmetric watermarking schemes, where the embedder and the detector use a different key to perform their tasks. On the contrary, non-disclosed information, which is to be kept secret, is denoted by $\mathcal{S}$. It follows that $\mathcal{S} = \{a, k_{\mathrm{D}}, k_{\mathrm{E}}\} - \mathcal{R}$. The game is a steady one *i.e.* the opponent is constrained to remain fair, if the information leakage concerning the secrets is small. Mathematically, it should be proven that:

$$I(\mathcal{O}; \mathcal{S}|\mathcal{R}) \sim 0 \tag{3}$$

where $I(\mathcal{O}; \mathcal{S}|\mathcal{R})$ is the mutual information between the observations and the secret information subject to the *a priori* knowledge the game assigns to the attacker. This quantity is important as it measures how the ignorance of the opponent about the secret decreases due to (or thanks to, according to the point of view) the observations. C.E. Shannon named this ignorance the equivocation [18]. It is given by :

$$H(\mathcal{S}|\mathcal{O}, \mathcal{R}) = H(\mathcal{S}|\mathcal{R}) - I(\mathcal{O}; \mathcal{S}|\mathcal{R}) \geq 0 \tag{4}$$

When the equivocation equals zero, the opponent has gathered enough observations to uniquely find the value of $\mathcal{S}$. C.E. Shannon speaks of a unicity distance [18]. Contrary to cryptography, the opponent of the watermarker usually does not need the exact value of the secret. If we assume that the secret is the watermark signal added to a content, a good estimation is usually sufficient to remove most of the watermark energy. For correlation-based detectors, even

Table 1
A framework for watermark security analysis. Each row differs according to the information the pirate has access to.

	A priori information			Pro's & Con's	
Scenario	a	k_{E}	k_{D}	Pirate's view	Owner's view
Security by obscurity	no	no	no	Need to focus on unfair attacks	Too much secret information
Symmetric watermarking	yes	no	no	Balance fair and unfair attacks	Difficulty in keeping k_{D} secret
Asymmetric watermarking	yes	no	yes	Better focus on fair attacks	Major threat from fair attacks
Playing open cards	yes	yes	yes	Very powerful fair attacks exist	Nothing but a dream ?

rough estimation can be used to forge a pirated content: the more accurate is the estimation, the less distortion is needed to hack the content [19].

When the above point of view is adopted, the classification given in table 1 is obtained. In the security-by-obscurity scenario, it is assumed that the attacker knows neither the algorithm used to embed and retrieve the watermark nor the embedding and detection keys. In such a scenario, the design of an effective fair attack may result to be a difficult task [7]. Hence it is better for the attacker to concentrate on un-fair attacks, since it is possible (likely) that some of the secret information leak out from the system thanks to the observations. Conversely, the watermarker has to put a significant effort to keep all the details about his system secret, a task which is extremely demanding.

By passing to the next row of the table, the common situation in which the watermarking algorithm is assumed to be known, but the embedding/detection keys are kept secret, is encountered. This is the scenario conventional watermarking algorithms refer to. In many applications, though, keeping the detection key secret may be a risky attempt. This is the case, for example, with applications where the detector is to be located in consumer electronics devices, *e.g.* with copy protection application [1]. Its disclosure is highly likely if the watermarking decoder is implemented in software on open platform like PCs [21], and still possible (yet, demanding a high technical level) even if implemented in hardware.

The difficulties with symmetric watermarking schemes where both the embed-

[7] Yet this is exactly what general purpose watermarking removal packages like Stirmark do [20].

ding and detection keys are kept secret, aroused the interest in asymmetric schemes, where the key used to retrieve the watermark is different from (a subset of) that used to embed it. Of course, in this way the effort necessary to protect secrets is significantly lower. At the same time, fair attacks represent a more and more insidious threat, up to a point that concerns exist on whether *robust* asymmetric watermarking will ever be possible.

Finally, an open-cards scenario may be conceived of where the attacker can access all the information he wants about both the embedder and the detector. Of course, in such a case, the effort put by the watermarker to protect system's secrets is minimized, thus forcing the attacker to rely only on fair attacks. On the other hand, in this case fair attacks may be extremely powerful, thus making the design of a secure, open-cards, watermarking systems extremely difficult (maybe impossible).

In the next sections, the four scenarios outlined above are discussed in more details, by considering both the watermarker's and attacker's points of view. The pro's and con's of the different approaches are highlighted and exemplified by the light of the current state of the art.

3 Security by Obscurity

This section explores the choice of relying on the fact that nothing is known by the attacker. Herein, nothing means neither the algorithms nor the tuning parameters are public *a priori* information.

3.1 The watermarker's side

This strategy was extremely common at the beginning of the digital watermarking history. The rationale was that if one cannot see the watermark, if one cannot fool the detector by any content transformation (because we deal herein with robust techniques), if one doesn't know how it is made, then, watermarking would definitively ensure security. The research efforts then focused on the robustness requirement, masking the need and even the concept of security in the watermarking community.

There are two shortcomings in this rationale. The first mistake is a misunderstanding of the functionality offered by a watermarking technique as already illustrated in subsection 1.1. The second mistake is the belief that a piece of information can remain secret, especially since it is an algorithm. There should be no need to say that if the security-by-obscurity strategy was chosen, then,

it would be neither possible to patent the watermarking technique nor to publish technical articles. It usually turns out that one discovers the company which provides its technique to the global system. The first thing an un-fair attacker does is to look for any piece of technical information coming from this company that would give him a clue on the used algorithms. Another method, which is more common in steganography than in watermarking, is to build statistical tests to discover what technique is used, *e.g.* in which domain the watermark signal has been added. The minimal amount of this *a posteriori* information that the adversary can gather is given by the instantiation of Eq. (3) when the secret to be disclosed is the algorithm and the observations are, at least, the watermarked content: $I(A; C_w = \mathrm{Emb}(C_o, A, M, K_{\mathrm{E}}))$. This threat of information leakages concerning the algorithms has really happened during the SDMI challenge [22]. Our conclusion is that, in watermarking, the algorithm can not remain secret. Hence, "obscurity" is not enough to enforce security against un-fair attackers.

In cryptography, people are more aware about these information leakages. Even in military applications, the motto is that an algorithm is disclosed within, on average, two years. In 1883, A. Kerckhoff wrote an article presenting the elementary cryptographic rules [23]. His main statement is that the designer of a cryptographic system must suppose that the adversary knows his algorithms in details except for a parameter called the secret key. Hence, the security of the cryptographic system only stems from storing the secret key in a safe place, the rest of the system being public. Kerckhoff's principle is a heuristic defended by two facts: there are proprietary algorithms (i.e., violating the Kerckhoff's principle) that have been hacked. The book of Singh gives numerous examples from the cryptographic field [24], the most famous being the hack of the Enigma encryption machine during the Second World War. Secondly, there are public encryption algorithms that remain unbroken (e.g. RSA, DES), even if weaknesses have been identified (for instance in key selection).

Kerckhoff's principle is more than a heuristic warning about the danger of the security-by-obscurity. It constitutes the basement of cryptanalysis, and hence the basement of security analysis of watermarking schemes. To get the basic idea across, we reflect the underlying concept of security level. This level is related to the amount of observation, the complexity, the amount of time, or the work as C.E. Shannon denoted it [18], that the attacker needs to gather in order to hack a system. What Kerckhoff means is that the watermarker should be aware of a lower bound of the security level. As the secrecy of an algorithm can not be fairly weighted, then, we should ignore it in security level estimations. This does not mean that obscurity is useless, it is just unproven security.

Having no clue about the watermarking technique, the fair attacker tries some content transformations to fool the watermark decoder. This is clearly a matter of robustness against intentional processing. We only concern ourselves with two issues in this section. The first one investigates how the pirate proceeds, the second one explores the risks he takes.

A huge part of the watermarking literature focuses on robustness, developing attacks and counter-attacks. At the beginning, a lot of research efforts dealt with compression, filtering or noise addition and, as a counter-measure, the selection of an embedding domain less sensitive to this transformation. One main idea is to slightly modify, via the use of a direct sequence spread spectrum communication scheme, the most perceptibly relevant parts of the content [25]. Nowadays, geometrical distortions are the main concern. Possible counter-measures are, for instance: to embed a template, an extra signal used to synchronize embedder and detector [26]; to embed the watermark signal in invariant domains [27]; to introduce redundancy in the watermark signal in order to reduce the space of potential delays; or to use use image self registration [28]. The watermarking community has also greatly benefited from some benchmark suites [29–31], entering in a virtuous circle of attacks and counter-attacks. Finally, the robustness of the watermarking techniques has been largely improved in the last years. To successfully hack protected contents, the attacker has to distort them down to a very low quality. Even if no certainty exists, and even if new, more powerful, attacks are invented nearly daily [32,33], we can realistically think, or simply assume, that in the end robustness issues will be given a satisfactorily answer, at least in the context of a given application scenario. This favors the strategy of the un-fair attacker disclosing the algorithms and the keys of the watermarking technique.

Some watermarkers name these kind of attacks *blind attacks*. Exploring this issue in more depth, we argue that this terminology is more appropriate in characterizing the pirate's state of mind rather than denoting a particular class of attacks. The fair attacker is in a blind state as he has no hint about the success of his attacks until his hacked contents are under the scrutiny of a watermark decoder. For a given attack, there are three types of watermarking techniques. The first ones are perfectly robust against this attack, meaning that their performance (such as the power of the detection test or the probability of correctly decoding the hidden message) is not affected. The second class gathers the techniques which are absolutely non-robust against this attack: it nullifies their performance. The last class is in the middle, when the attack lowers the performance to a given extent. For instance, if the power of the detection test decreases down to $1/2$, the attack succeeds in average on one out of two contents. Too much papers claim their proposed watermarking

technique is robust to, for instance, jpeg compression as tested on the 'Lena' image. But the only relevant experiment, in watermarking detection, is to plot the power of the test against the quality factor of the compression.

Moreover, the fair attacker in a blind state has no clue about the type of techniques he faces. For some applications, this is not a matter. For example, in DVD copy protection, a consumer electronics device has a watermark detector which prevents from recording protected contents. The pirate can do whatever experiment secretly and safely at home. In other scenarios, he has no access to a watermark decoder, and, a failure in the attack is a dead-lock. The cost of a failure might be prohibitive. This is often the case in the professional domain, where, once caught, the attacker would be sued and ruined by trials.

To conclude this section, it turns out that security-by-obscurity is not a steady state of the table 1. The easiest path is to go for an un-fair attack, trying to jump into the next row of the table. On the other hand, the watermarker is advised by Kerckhoff's principle to not to ensure security by relying on obscurity only.

4 Symmetric watermarking

As we have seen, it is not reasonable to assume that the watermarking algorithm remains unknown; as a first step, thus, we make the hypothesis that solely the embedding and decoding keys are kept secret. Under such an hypothesis the fair attacks try to remove or make unreadable the watermark based solely on the *a priori* information about the embedding and decoding algorithms, while the unfair try to discover the secret keys, and, based on this *a posteriori* information, remove or make unreadable the watermark. The latter class of attacks is more difficult, but also more effective in achieving its goal, in the sense that the amount of attack distortion needed is surely lower.

We now analyze which possibilities exist and the challenges that the watermarker and the pirate must undertake.

4.1 The watermarker's side

In general, the watermarker worries about two issues. Firstly, he has to select a watermarking technique which is robust to the content transformations the attacker may resort to in the current application. We already discussed this point in subsection 3.2. We would like to insist here on the fact that the sentence "this technique is robust" does not make sense in general. We believe,

in fact, that for almost all applications it is possible to find a watermarking technique which is robust to the content transformations allowed in that particular context. For instance, in video watermarking, a rotation of the frames of more than few degrees is a moot attack: will people accept to turn their head to watch these hacked movies?

This argument must be tempered with the following fact. Knowing the algorithm, the pirate can resort to more powerful attacks, since he is able to play in the embedding domain. In other words, he has access to feature vectors $\vec{f}$ because the extraction function is now public. More sophisticated attacks than classical content transformations rely on noise removal filtering, in particular if suitable statistical models of the original features and of the watermark signal are available. For example, S.Voloshynovskiy and *al.* developed a watermark removal filter based on Maximum Likelihood or Maximum A Posteriori probability criteria [34]. In practice, the pirate looks for the best approximation of the original document, by assuming that the watermark can be viewed as disturbing noise. Similarly, Wiener filtering can be adopted to try to separate the watermark and the host document. The theoretical issue behind this is whether a perfect mixing Mix(.) of the original features and the watermark signal is possible [9].

A possible countermeasure, suggested by J. Su and *al.* is to follow the Power Spectrum Condition [35] stating that the power spectrum density of the watermark should be shaped like the one of the feature vector: $S_{\vec{w}} \propto S_{\vec{f}_o}$. Another possibility, proposed by Le Guelvouit and *al.*, is to embed the watermark signal and then to self attack the resulting signals $\vec{f}_w$ by a Wiener filtering [36]. This highly diminishes the efficiency of watermark removal filters.

Additionally, for watermarking systems implemented in public consumer electronics devices, the watermarker must ensure that these devices can not be forced. If there is no place where the keys are stored safely, then, security is impossible with the rule of this section. This precaution has a price: software for PC is non-secure but cheap compared to expensive secure processing units for smart cards. Secure implementation of security primitives is a real art which is out the of scope of this article.

4.2 The attacker's side

In this scenario, the attacker is not blind. Because we assume robust watermarking, he knows that the watermarker did a great job: the hack of protected contents at a distortion below a perceptual bound (this one depends on the application) is hard to find. This certainly ruins his business plan as nobody is interested in such heavily distorted hacked contents.

His strategy is to refuse the rule of the game. The disclosure of the secret keys is the mean to forge pirated contents at a low quality loss. To get the basic idea across, the key will allow him to decode the hidden message. Having in his hands the key, the data to be embedded and the watermarked content, which is a close version of the original content, the synthesis of the watermark signal and its subtraction from the watermarked content are considerably easier. For example, with classical spread spectrum watermarking the attacker may act as follows:

$$c_p = c_w - (\mathrm{Emb}(c_w, a, \hat{m}, k_\mathrm{E}) - c_w) \sim c_o, \tag{5}$$

where c_p is the pirated content. Note that such a simple attack does not work with quantization index modulation schemes, since in that case the embedded signal depends on the host signal, in such a way that $\mathrm{Emb}(c_w, a, \hat{m}, k_\mathrm{E}) = c_w$. However, a similar algorithm for quantization index modulation schemes has also been developed [37].

As the estimation of the secrets is based on the observations, the four types of attacks of subsection 1.3 are now toured. From now on, the attacks are two-step processes:

(1) Learning phase: Observe $\mathcal{O}$ to gain knowledge from them.
(2) Practice phase: Use this knowledge to hack the targeted contents.

4.2.1 Only watermarked content type

In this scenario, the information leakage is measured by Eq. (6). In this case:

$$I(\{\vec{F}_{w,i}\}; K_\mathrm{E}|A) = I(\{\mathrm{Emb}(\vec{F}_{o,i}, M_i, K_\mathrm{E})\}; K_\mathrm{E}) \tag{6}$$

where conditioning to A has been removed for simplicity in the mutual information. Feature vectors replaced contents because the opponent has access to the embedding domain. $\{\vec{F}_{o,i}, M_i\}$ are sources of entropy, whereas K_E has been fixed by the watermarker before the beginning of the game (still denoted in capitals as the mutual information is an integral over the set of keys).

As an illustration, we take the very much simple case where $M_i = 1, \forall i \in \mathcal{I}$, and algorithm a is a Direct Sequence Spread Spectrum (DS-SS) based watermarking technique. There is, then, a bijection between k_E and $\vec{w}$. Indeed, the pirate is usually more interested in $\vec{w}$ than in k_E, if his goal is to forge unwatermarked contents. With the simple assumption that $\vec{F}_o$ and $\vec{W}$ represent independent gaussian random vectors whose covariance matrices are $\mathbf{R}_o$ and $\mathbf{R}_w$, then, the leakage of information from one observed watermarked content equals:

$$I(\mathrm{Emb}(\vec{F}_o, 1, \vec{W}); \vec{W}) = H(\vec{F}_w) - H(\vec{F}_w|\vec{W}) \tag{7}$$
$$= H(\vec{F}_w) - H(\vec{F}_w - \vec{W}|\vec{W}) \tag{8}$$
$$= H(\vec{F}_w) - H(\vec{F}_o) = \frac{1}{2}\log\left(1 + \frac{\det \mathbf{R}_w}{\det \mathbf{R}_o}\right) \tag{9}$$

This quantity is minimized for $\mathbf{R}_w = P\mathbf{R}_o/\sigma_o^2$, P being the power of the watermark vector. In other words, the Power Spectrum Condition minimizes the information leakage.

Under this condition, the work of the adversary in order to have an accurate estimation of $\vec{w}$ is to gather and average $O(\sigma_o^2/P)$ watermarked contents (we assume these represent independent signals). Note that this averaging process can be done with raw contents, but it achieves higher efficiency when done on feature vectors, especially if signal estimation is used as detailed in section 4.1 [10].

Things are more complex as message entropy gets higher. The watermark signal then depends on k_E but also on random variable M, via the modulation scheme Gen(.). But, it should be still possible to estimate the subspace where the watermark signal spans. This is especially true when a spread transform is used to increase the watermark signal to noise ratio. The message is hidden modifying the projection of vector $\vec{F}_o$ onto several secret carriers $\{\vec{p}_k\}_{k\in\mathcal{L}}$. Hence, vector $\vec{W}$ lives in the small $|\mathcal{L}|$-dimension subspace $\mathrm{Vect}(\{\vec{p}_k\}_{k\in\mathcal{L}})$ whereas vectors $\vec{F}_o$ span over $\mathbb{R}^N$. As far as we are concerned, we do not know any research work on this subject.

The only-watermarked-content-type also encloses the very special case of the tracing application by fingerprinting where $\mathcal{O} = \{\vec{f}_{w,i} = \mathrm{Emb}(\vec{f}_o, m_i, k_\mathrm{E})\}$. The average attack is then called a collusion (a term from the cryptography community). Its goal is not to discover secrets but to directly produce an un-traceable content $\hat{c_o}$. Anti-collusion codes [15–17] have been devised as a counter-measure. The basic idea behind them is that different codes should have at least a part in common: it should not be always the same part, but given any subset of the whole set of codes, the codes belonging to the subset should have a common part. When some watermarked feature vectors are averaged, the common parts of the codes do not reduce their strength, thus making possible to at least identify a subset of the colluders.

Things are much more complex when an informed embedding approach is used, since in this case the watermark signal depends on two sources of entropy M and $\vec{F}_o$.

4.2.2 Chosen watermarked content type

In this case, the opponent is assumed to have access to a watermark decoder. A possibility is to iteratively modify the document until the decoder is no longer able to recover the watermark. This fair approach has a strong limitation, in that, being the modifications performed almost randomly, the time to find a successful hack for one content within a low quality loss is not deterministic and, possibly, very high. The unfair version has been, on the contrary, demonstrated to be very effective [38] with the so called sensitivity attack. The learning phase results in the estimation of the boundary of the decision region, *i.e.* the *locus* separating the region of feature vectors considered as watermarked by the decoder, and the region that, on the contrary, is considered as non-watermarked. In the practice phase, removal is easily performed by looking for the boundary point which is nearest to the watermarked content (closest point attack [39]).

In the learning phase, the boundary is estimated as follows: starting from the watermarked features $\vec{f}_{w,i}$, they are iteratively modified until the watermark presence is no longer detected. For instance, knowing that the null vector belongs to the non-watermarked region, the opponent is sure to find such a a feature point $\vec{f}_{\mathcal{B},i}$ near to the boundary in iteratively reducing the energy of $\vec{f}_{w,i}$.

This feature point is now corrupted by adding random vectors $\vec{n}_j$, and the response of the detector measured for each corrupted vector $\vec{f}_{\mathcal{B},i} + \vec{n}_j$. This leads to the estimation of the local orthogonal vector $\vec{u}_i^{\perp}$ of the boundary. T. Kalker has proven that [38]:

$$I(\vec{U}_i^{\perp}; \{\hat{M}_{i,j}, \vec{F}_{\mathcal{B},i} + \vec{N}_j\}_{j \in \mathcal{JC}}) \propto |\mathcal{JC}| \quad \text{for } |\mathcal{JC}| \ll N \tag{10}$$

The ignorance of the opponent about the value $\vec{u}_i^{\perp}$ decreases linearly with the figure of trials, at least at the beginning of the experiment. It is clear that as the equivocation of Eq. (4) goes to zero, the number of observations increasing, the mutual information tends to zero. No law has been shown for $|\mathcal{JC}| \lesssim N$ up to now. Moreover, if we assume that the sign of the components of vector $\vec{u}_i^{\perp}$ yields enough information to the hacker, then, the equivocation starts at $H(\text{sign}(\vec{u}_i^{\perp})) = N$ in bits. An estimation of $\text{sign}(\vec{u}_i^{\perp})$ requires then $O(N)$ decodings, as shown in figure 2. For a correlation based detector using a fixed threshold as in DS-SS schemes, the boundary is an hyperplane and the orthogonal direction at any point allows to recover the whole boundary. The security level of this scheme against a chosen watermarked content attack is at most $O(N)$. For more complex detection regions, the orthogonal direction only gives information about the locally tangent hyperplane at location $\vec{f}_{\mathcal{B},i}$. The same estimation should then be repeated for other boundary points. Usually, the boundary is a parametric surface, so that a finite figure of tangent

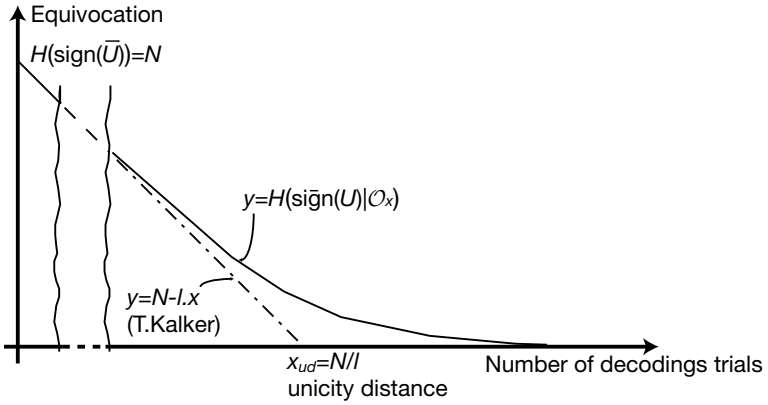

Fig. 2. Graph of the equivocation against the number of decoding trials.

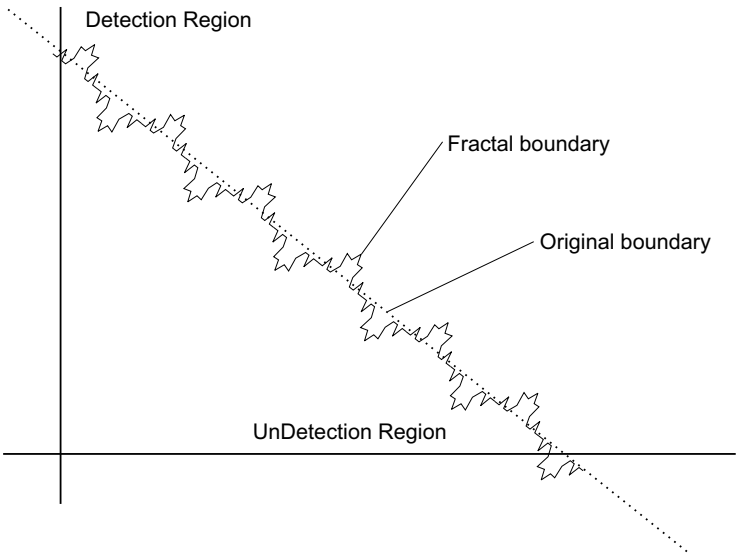

Fig. 3. Example of a not parameterizable boundary in the feature space.

hyperplanes is enough to estimate these parameters. It has been proven that N boundary points are necessary to estimate the parameters of schemes using second order statisitc based detector (e.g. asymmetric schemes [40] and JANIS [41]). Then, the security level is in the order of $O(N^2)$ decoder trials.

A ultimate countermeasure against the sensitivity attack has been recently proposed [42]. The success of the sensitivity attack is due to the fact that the boundary is a parametric curve. The idea in this case is to modify the detection boundary in such a way to make it not parameterizable by using a fractal curve. This solution is illustrated in Figure 3, where the original boundary, here a simple hyperplane (dotted line), is modified to become a Peano curve. There is no way to find locally tangent hyerplane based solely on the detector response. The embedding procedure should be, of course, also modified, to be sure that the watermarked document actually lies in the detection region.

4.2.3 Original and watermarked pairs type

The case of having, at the attacker disposal, original and watermarked contents pairs (either chosen or not) is straightforward both in the case of fair and unfair attacks. The fair attack has not even sense, given that the original unwatermarked content is available. The unfair one can be easily implemented by comparing a pair of documents and inverting the embedding rule (which is known) in such a way to recover the embedded watermark signal. The mixing function can be broken, and $\vec{w}$ estimated almost perfectly. In the simple example described in subsection 4.2.1, one pair of contents is enough to disclose the secret signal.

There are two possible countermeasures against pair attacks. The first one makes the watermarking signal strongly content-dependent, so that, once recovered, the watermark can not be used for removing it from (or adding it to) other documents. In this framework, a possibility consists in making $\vec{w}$ dependent, further than on the secret and a message, on a robust hash value of the content [43]. With such an approach, the design of the hashing function may be very difficult, given that it should be robust (*i.e.* should produce the same value with moderately modified copies of the document) and theoretically difficult to invert. Another option uses side informed watermark embedding so that, by nature, $\vec{w} = \mathrm{Gen}(\vec{f_o}, m, k_\mathrm{E})$ [37,41]. The security level is not *a priori* increased because these functions are mainly designed to improve capacity or probability of good detection, and not to provide security. For the JANIS scheme with a second order detector, and $m_i = 1, \forall i \in \mathcal{IO}$, the security level is upper bounded by $O(N)$ pairs of independent vectors [10].

A second countermeasure randomizes the watermark signal: $\vec{w} = \mathrm{Gen}(m, r, k_\mathrm{E})$ where r is a random value changing at each embedding [40]. The security level is then upper-bounded by $O(|\mathcal{K}_\mathrm{E}|)$ tries of keys, i.e. the complexity of a brute force attack for schemes where $m_i = 1, \quad \forall i \in \mathcal{IO}$.

4.3 Conclusion concerning this case

When compared to the "security by obscurity" scenario, the analysis developed in this section, represents a dramatic improvement, since it is now possible to quantify the security level of a given watermarking system. The use of information theory to quantify information leakages is not new in security. It dates back to Shannon's cryptographic article [18]. Its use in data hiding is very rare with known exceptions [44,45,10].

Note that the secrecy of the key has been analyzed herein, but the opponent might be interested in reading (or writing) hidden messages, rather than disclosing the key. In other words, the opponent seeks to obliterate the wa-

termarking channel (which resembles the physical layer of a communication system) in our analysis, but other threats are the unauthorized access to this channel. Encryption of the hidden messages prevents from disclosing their semantic sense if the opponent succeeds to decode the hidden bits. Digital signature of the hidden messages prevents from usurping the right to write onto this channel. Of course, these tools do not protect the sustaining of the watermark signal. Furthermore, the use of these cryptographic primitives on top of the robust watermarking channel may not be possible as its capacity is usually very low. To give an order of magnitude, texts are usually encrypted by blocks of 128 bits (AES) or 1024 bits (RSA), and the recommended sizes of digital signatures are 1024 bits (DSA) or 320 bits (elliptic curves signatures). As an alternative, signal processing tools might provide these primitives. The invention of a modulation scheme such as $I(\vec{W} = \mathrm{Gen}(M, K_{\mathrm{E}}); M) = 0$ is a real challenge. A very good paper with respect to this subject, which is also, with Cachin's article [44], the pioneer work about security of watermarking, was written by T. Mittelholzer in 1999 [45].

5 Asymmetric watermarking: towards public key detection

In this section, algorithm a and detection key k_{D} are public data. Before analyzing this rule, the need of public key detection is justified. In copyright protection, the decoding of a watermark in a work might bring a proof of ownership to a doubtful person. It implies that this *a priori* non-trusted person has access to the decoder, and he could steal the decoding key. Protocols mixing zero-knowledge disclosure cryptography and watermarking are very promising [7]. But, they need a bidirectional communication between the prover and the verifier. In copy protection, the decoder embedded in consumer electronics devices is also in a hostile environment, and a bi-directional link between the producer stage and the device may be difficult or just impossible to establish. The idea of a public detection key is very compelling because secure electronic chips are very expensive. Security, then, resides only on the secrecy of the embedding key.

This concept is quite astonishing, but it is a reality in cryptography. Digital signatures are based on asymmetry: a verifier checks with the related public key what have been signed by a private key. Is this possible in watermarking? As far as the authors know, the answer can not yet be given. By following the analogy with cryptography, the embedding and the detection processes must be made asymmetric, in relying on different keys (or sets of parameters). This is why public-detection watermarking is usually referred to as asymmetric watermarking. However, as we will see below, key asymmetry is by no means sufficient to provide security in a public-detection environment. At least, the watermarker must have a proof that the knowledge of the detection key does

not bring a full or partial *a priori* information about the embedding key.

5.1 The attacker's side

In order to illustrate why key asymmetry does not always help the design of a public-detection watermarking scheme, let us consider the example of the randomized embedding process described in section 4.2. Denote $k'_{\mathrm{E}} = \{k_{\mathrm{E}}, r\}$. The watermarking system with keys k'_{E} and k_{D} is asymmetric as r is a random only known at the embedding side. Thus, knowing k_{D}, there is no way to disclose k'_{E}. Yet, asymmetry does not imply robust public key watermarking. All known asymmetric schemes are hacked so far when k_{D} is public [8]: paper [46] explains an attack valid for almost all known asymmetric schemes. In other words, whereas asymmetry is certainly helpful, it is far from being sufficient to ensure security in a public detection scenario.

5.2 The watermarker's side

The reason why asymmetric schemes are not (up to now) secure public key detection methods is the following. The knowledge of the algorithm and the detection key implies the knowledge of the boundary of the detection region. The closest point attack is then a deadlock. Will there be a solution?

Here are some hand-waving justifications about a theoretical watermarking detection avoiding this pitfall. The detection process must be an algorithm which is a binary test: $\mathrm{Dec}(.) : \mathbb{R}^N \rightarrow \{0, 1\}$. It is an indicator function of the detection region. But, it must not reveal the boundary of this region to prevent the closest point attack. Does this kind of mathematical function exist? Fractal functions pertain to this principle. We do not know how to build a watermarking scheme from this type of functions, but, at least a mathematical tool yields the good property required for public key detection watermarking. Such a detection process does not need to be private. The pirate must test every point of the space in order to disclose the boundary. The watermarker wins the game if this brute force attack lasts an exponential (with respect to N) amount of time. This is seemingly possible: imagine the pirate quantizes the real axis into B bins, then the map of $\mathbb{R}^N$ is a grid containing B^N points to be tested. The watermark must prove that there is no other way to disclose the boundary. The next issue is how the knowledge of the embedding key allows him to watermark contents in a polynomial amount of time.

[8] Yet, they are not useless, because they provide higher security levels than classical schemes when the detection key is secret as already mentioned in 4.2.

6 Playing open cards

The path followed in the previous sections has been one of progressively diminishing the information the owner must keep secret. In this way the effort needed to face with unfair attacks diminishes, while leaving the owner's side open to more and more powerful fair attacks. When brought to its extreme consequences, this process leads to a situation in which the attacker can access all the information he desires, i.e. he knows the watermarking algorithm, as well as both the embedding and detection keys used by the system. In this section, we briefly review the challenges set to the watermarker and the attacker by this, so to say, open cards scenario.

6.1 Watermarker's side

It is obvious that the open cards scenario is the most favorable one from the point of view of unfair attacks. According to our scheme, in fact, no care has to be taken to keep information secret, simply because the watermarking algorithm assumes that the attacker can obtain all the information he desires. This may also be considered as a pessimistic scenario where the watermarker assumes that no defense is possible against unfair attacks, since, in the end, the attacker will also manage to unreveal the watermarker's secrets. Playing an open cards game is a possibility to get around the problem. An obvious question is whether in this case robustness against fair attacks is possible.

In order to get more insight into the security problems set by the open cards scenario, let us consider in more details the reasons that led us to consider asymmetric watermarking schemes. As we have seen in subsection 4.2, one of the most powerful attacks that can be conceived against virtually any watermarking scheme is the closest point attack. In its fair version, such an attack assume that the attacker knows the boundary of the detection region, and hence can make the watermark unreadable by simply moving the host data to the closest point of the non-detection region. The unfair version of the closest-point attack only differs from the fair version, in that the boundary of the detection region is estimated by the attacker through a trial and error procedure. The reason why asymmetric watermarking schemes may provide a solution to the closest point attack can be explained as follows. Let us assume that the detection region is described in such a complicated way that moving a point inside and outside it, while matching the invisibility constraint, is a computationally unfeasible operation. This may be the case, for example, with the boundary depicted in figure 3. Knowing the detection boundary now does not help the attacker in any way. Unfortunately, having such a complicated detection region also makes watermark embedding a very difficult task. Asym-

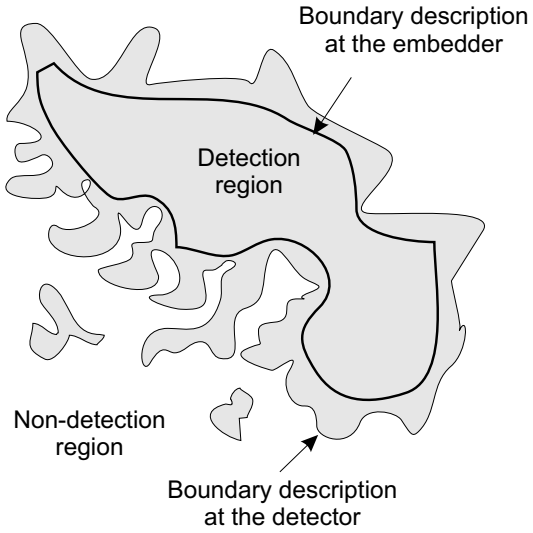

Fig. 4. Asymmetric watermarking. A simplified description of the detection region is available at the embedder.

metric watermarking may help in solving this apparent deadlock by letting the embedder know an alternative, hopefully simpler, description of the detection region. With such a description in his hands, the embedder can easily move any point in the host feature space inside the detection region, while, at the same time, satisfying the invisibility constraint. Such a situation is exemplified in figure 4, where the complex detection region boundary available at the detector is depicted together with a smoother boundary used by the embedder to watermark the to-be-protected content. According to this scenario, the necessary asymmetry between watermark embedding and watermark removal (respectively corresponding to entering and exiting the detection region) is obtained by assigning the embedder and the detector (and hence the attacker) a different description of the detection region, i.e. a different key. One may wonder whether such an asymmetry may be obtained directly by properly designing the detection region. In other words, would it be possible to design the detection region in such a way that it is easy to move a point inside it, but very difficult to bring a point outside it, by matching, at the same, the imperceptibility constraint ? If this is possible, there is no need to distinguish between the embedding and detection keys, and to keep any of these keys secret: the watermarker can play open cards.

No answer has been given to the above questions so far, even if the design of a detection region such as the one described above immediately appears to be a very difficult task. However, until an explicit proof that a detection region with these characteristics can not be built, the possibility of developing a secure watermarking system following the open card approach can not be ignored. A more detailed discussion of the open cards approach, can be found in [39], along with some hints on how an asymmetric detection region could be built .

6.2 Pirate's side

From the pirate's point of view, the open-cards scenario may be a favorable one, since in this case very powerful fair attacks can be conceived of. However, if the possibility of building an intrinsically asymmetric detection region with the characteristics described in the previous section is proved, the pirate's task becomes hopeless, since no further possibility exists for him to resort to unfair attacks.

7 Conclusion

Aiming at discussing watermarking security from as wider a perspective as possible, we introduced a framework in which each watermarking system is classified according to the knowledge in the hands of a possible attacker. In order to further clarify the nature of such a knowledge, we introduced the concept of fair and unfair attacks. In this way we managed to distinguish between the information which is publicly available to the pirate and that gained by the pirate through a set of attacks, expressly designed to disclose watermarker's secrets.

With this general formulation in mind, we selected the four scenarios summarized in table 1. Whereas it is widely agreed that system designers should not resort to the security-by-obscurity scenario, and the open-cards approach may seem to be unrealistic, the choice between symmetric and asymmetric watermarking is still a current research issue.

As a second contribution, we described a mathematical framework, inspired by information theory principles, whereby the security of any watermarking algorithm with respect to unfair attacks can be quantified. This represents a significant improvement with respect to the current state of the art, according to which security is often treated at a very empirical level.

As a final remark, we would like to stress out that, whereas cryptographic-like security may be out-of-reach for watermarking systems, and maybe not even required in most cases, it is still important that a significant effort is made in order to, at least, define and quantify watermarking security in a precise and solid mathematical sense.

References

[1] J. Bloom, I.J. Cox, T. Kalker, J.-P. Linnartz, M.L. Miller, C.B.S. Traw, Copy protection for DVD video, Proceedings of the IEEE 87 (7) (1999) 1267–1276, special issue on identification and protection of multimedia information.

[2] M. Maes, T. Kalker, J.-P. Linnartz, Joop Talstra, Geert Depovere, Jaap Haitsma, Digital watermarking for DVD video copy protection, Signal Processing Magazine 17 (5).

[3] Copy protection technical working group, `http://www.cptwg.org`.

[4] SDMI, Call for proposal for phase ii screening technology, version 1.0 (Feb. 2000).

[5] S. Craver, N. Memon, B.-L. Yeo, M.M. Yeung, Resolving rightful ownership with invisible watermarking techniques: limitations, attacks, and implications, IEEE Journal of selected areas in communications 16 (4) (1998) 573–87, special issue on copyright and privacy protection.

[6] J. Stern, J.-P. Tillich, Automatic detection of a watermarked document using a private key, in: I. S. Moskowitz (Ed.), 4th Int. Work. on Information Hiding, Vol. 2137 of Lecture Notes in Computer Science, Springer, Pittsburgh, PA, USA, 2001, p. electronic version.

[7] A. Adelsbach, S. Katzenbeisser, A. Sadeghi, Cryptography meets watermarking: Detecting watermarks with minimal or zero knowledge disclosure, in: Proc. of European Signal Processing Conference, Toulouse, France, 2002.

[8] T. Kalker, Considerations on watermarking seccurity, in: Proc of the IEEE Multimedia Signal Processing MMSP'01 workshop, Cannes, France, 2001, pp. 201–206.

[9] T. Furon, J. Oostven, J. Van Bruggen, Security analysis, Deliverable D.5.5, CERTIMARK IST European Project (2002).

[10] T. Furon, Use of watermarking techniques for copy protection, Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications. (2002).

[11] W. Diffie, M. Hellman, New directions in cryptography, IEEE Trans. on Information Theory 22 (6) (1976) 644–54.

[12] I. J. Cox, M. L. Miller, A. L. McKellips, Watermarking as communications with side information, Proc. IEEE 87 (7) (1999) 1127–1141.

[13] I. Cox, J.-P. Linnartz, Some general methods for tampering with watermarks, IEEE Journal on selected areas in communications 16 (4) (1998) 587–93, special issue on copyright and privacy protection.

[14] A. Herrigel, S. Voloshynovskiy, Y. Rystar, The watermark template attack, in: P.W. Wong, E. Delp (Eds.), Security and Watermarking of Multimedia Contents III, SPIE Proceedings, San Jose, Cal., USA, 2001.

[15] D. Boneh, J. Shaw, Collusion-secure fingerprinting for digital data, IEEE Tran. on Information Theory 44 (5) (1998) 1897–1905.

[16] J. Dittmann, A. Behr, M. Stabenau, P. Schmitt, J. Schwenk, J. Ueberberg, Combining digital watermarks and collusion secure fingerprints for digital images, in: P.W. Wong, E. Delp (Eds.), Security and Watermarking of Multimedia Contents, SPIE Proceedings, San Jose, Cal., USA, 1999.

[17] W. Trappe, M. Wu, K. J. R. Liu, Joint coding and embedding for collusion-resistant fingerprinting, in: Proc. European Signal Processing Conf. EUSIPCO 2002, Toulouse, FR, 2002.

[18] C. Shannon, Communication theory of secrecy systems, Bell system technical journal 28 (1949) 656–715.

[19] G. Langelaar, R.L. Lagendijk, J. Biemond, Removing spatial spread spectrum watermarks by non-linear filtering, in: Proc. of 9th European Signal Processing Conference, EUSIPCO, Island of Rhodes,Greece, 1998.

[20] F. Petitcolas, Stirmark, `http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark`.

[21] A. Patrizio, DVD privacy: It can be done, `http://www.wired.com/news/technology/1,1282,32249,00.html`.

[22] S. Craver, J. Stern, Lessons learned from SDMI, in: Proc of the IEEE Multimedia Signal Processing MMSP'01 workshop, IEEE, Cannes, France, 2001.

[23] A. Kerckhoffs, La cryptographie militaire, Journal des sciences militaires 9 (1883) 5–38.

[24] S. Singh, The code book, Fourth Estate Limited, 1999.

[25] I. Cox, J. Kilian, T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Trans. on Image Processing 6 (12) (1997) 1673–1687.

[26] S. Pereira, T. Pun, Fast robust template matching for affine resistant image watermarks, in: A. Pfitzmann (Ed.), Proc. of the third Int. Workshop on Information Hiding, Springer Verlag, Dresden, Germany, 1999, pp. 199–210.

[27] J. O'Ruanaidh, T. Pun, Rotation, scale and translation invariant spread spectrum digital image watermarking, Signal Processing, Special issue on copyrigth protection and control 66 (3) (1998) 303–17.

[28] P. Bas, B. Macq, A new video-object watermarking scheme robust to object manipulation, in: Proc. of Int. Conf. on Image Processing, IEEE, Thessaloniki, Greece, 2001.

[29] F. A. P. Petitcolas, Watermarking schemes evaluation, IEEE Signal Processing 17 (5) (2000) 58–64.

[30] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Maillet, T. Pun, Second generation benchmarking and application oriented evaluation, in: Springer-Verlag (Ed.), Fourth workshop on information hiding, Lecture Notes in Computer Science, Pittsburgh, PA, USA, 2001.

[31] V. Solachidis, A. Tefas, N. Nikolaidis, S. Tsekeridou, A. Nikolaidis, I.Pitas, A benchmarking protocol for watermarking methods, in: Proc. 2001 IEEE Int. Conf. on Image Processing - ICIP'01, Thessaloniki, GR, 2001, pp. 1023–1026.

[32] F. A. P. Petitcolas, D. Kirovski, Blind pattern matching attack on audio watermarking systems, in: Proc. of the IEEE Int. Conf. on Acoustic, Speech and Signal Processing, Orlando, Florida, USA, 2002.

[33] C. Rey, G. Dorr, J.-L. Dugelay, G. K. Csurka, Toward generic image dewatermarking?, in: Proc. of the IEEE Int. Conf. on Image Processing, Vol. 3, Rochester, NY, USA, 2002, pp. 633–636.

[34] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, T. Pun, A stochastic approach to content adaptive digital image watermarking, in: D. Pfitzman (Ed.), Third workshop on information hiding, Lecture Notes in Computer Science, Dresden, DE, 1999.

[35] J. Su, J. Eggers, B. Girod, Analysis of digital watermarks subjected to optimum linear filtering and additive noise, Signal processing 81 (2001) 1141–1175.

[36] S. Pateux, G. Le Guelvouit, C. Guillemot, Perceptual watermarking of non i.i.d. signals based on wide spread spectrum using side information, in: Proc. of IEEE Int. Conf. on Image Processing, Vol. 3, Rochester, NY, USA, 2002.

[37] J. Eggers, B. Girod, Informed Watermarking, Kluwer Academic Publishers, 2002.

[38] T. Kalker, A security risk for publicly available watermark detectors, in: Benelux Information Theory Symposium, 1998, veldhoven, The Netherlands.

[39] M. L. Miller, Is asymmetric watermarking necessary or sufficient?, in: Proc. European Signal Processing Conference - EUSIPCO 2002, Toulouse, FR, 2002.

[40] T. Furon, P. Duhamel, An asymmetric watermarking method, accepted to the special issue on signal processing for data hiding in digital media & secure content delivery, IEEE Trans. on Signal Processing (March 2003).

[41] T. Furon, G. Silvestre, N. Hurley, JANIS: Just Another N-order side-Informed Scheme, in: Proc. of Int. Conf. on Image Processing ICIP'02, Rochester, NY, USA, 2002.

[42] M. F. Mansour, A. H. Tewfik, Secure detection of public watermarks with fractal decision boundaries, in: Proc. European Signal Processing Conference - EUSIPCO 2002, Toulouse, FR, 2002.

[43] G. Depovere, T. Kalker, Secret key watermarking with changing keys, in: Proc. of Int. Conf. on Image Processing, Vol. 1, IEEE, Vancouver, Canada, 2000, pp. 427–429.

[44] C. Cachin, An information-theoretic model for steganography, in: D. Aucsmith (Ed.), Proc. of the second Int. Workshop on Information Hiding, Vol. 1525 of Lecture Notes in Computer Science, Springer Verlag, Portland, Oregon, U.S.A., 1998, pp. 306–318.

[45] T. Mittelholzer, An infomation-theoritic approach to steganography and watermarking, in: A. Pfitzmann (Ed.), Proc. of the third Int. Workshop on Information Hiding, Springer Verlag, Dresden, Germany, 1999, pp. 1–17.

[46] T. Furon, I. Venturini, P. Duhamel, Unified approach of asymmetric watermarking schemes, in: P.W. Wong, E. Delp (Eds.), Security and Watermarking of Multimedia Contents III, SPIE, San Jose, Cal., USA, 2001.