The Return of the Sensitivity Attack

Pedro Comesaña, Luis Pérez-Freire and Fernando Pérez-González *

Signal Theory and Communications Department
University of Vigo, Vigo 36310, Spain
{pcomesan, lpfreire, fperez}@gts.tsc.uvigo.es

Abstract. The sensitivity attack is considered as a serious threat to the security of spread-spectrum-based schemes, since it provides a practical method of removing watermarks with minimum attacking distortion. This paper is intended as a tutorial on this problem, presenting an overview of previous research and introducing a new method based on a general formulation. This new method does not require any knowledge about the detection function nor any other system parameter, but just the binary output of the detector, being suitable for attacking most known watermarking methods. Finally, the soundness of this new approach is tested by attacking several of those methods.

1 Introduction

In its early years, digital watermarking was conceived as a solution to the problems of illegal copy control and intellectual property rights (IPR) protection. Perhaps for this reason and the analogies commonly made to the field of cryptography, watermarking was declared as synonymous to security [1]. However, watermarking research until now has much more to do with *robustness* than with *security*: roughly speaking, watermarking security [2] may be related to attacks which try to gain knowledge about certain secret parameters of the watermarking system, whereas robustness is more concerned with attacks whose aim is to degrade the performance of the watermarking system.

In watermarking for IPR protection and copy control, the aim is to distinguish whether the digital media at hand contains a certain watermark or not. This problem is known as *watermark detection*,[1] and is commonly modeled as a binary hypothesis testing problem. In a general setup, the watermarking of a

* This work was partially funded by *Xunta de Galicia* under projects PGIDT04 TIC322013PR and PGIDT04 PXIC32202PM; MEC project DIPSTICK, reference TEC2004-02551/TCM; FIS project IM3, reference G03/185 and European Comission through the IST Programme under Contract IST-2002-507932 ECRYPT. ECRYPT disclaimer: The information in this paper is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

[1] Watermark detection and *watermark decoding* must be regarded as different problems, since in the latter (which is often referred to as *data hiding*) the objective is to decode the embedded message .

digital document $\mathbf{x}$, which is arranged as a column vector of dimension n, can be expressed as $\mathbf{y} = \mathbf{x} + \mathbf{w}$, with $\mathbf{w}$ the watermark. Hence, the hypothesis testing problem can be written as

$$\begin{array}{l} H_0 : \mathbf{y} = \mathbf{x} \\ H_1 : \mathbf{y} = \mathbf{x} + \mathbf{w} \end{array}.$$

In detection, we must adapt this test to take into account that the watermarked signal could have been attacked; this attack will be modeled as the addition of a vector $\mathbf{t}$, yielding a signal $\mathbf{z} = \mathbf{y} + \mathbf{t}$. Note that $\mathbf{w}$ may be made key-dependent in order to improve the security of the system. The optimal solution to the hypothesis test is given by the likelihood ratio test, i.e.,

$$l(\mathbf{z}) = \frac{f_{\mathbf{Z}|H_1}(\mathbf{z}|H_1)}{f_{\mathbf{Z}|H_0}(\mathbf{z}|H_0)} \begin{array}{c} H_1 \\ > \\ \leq \\ H_0 \end{array} \eta, \tag{1}$$

where $f_{\mathbf{Z}|H_i}(\mathbf{z}|H_i)$ is the pdf of $\mathbf{Z}$ conditioned to hypothesis H_i and η is a threshold which can be adjusted so as to optimize a certain criterion (Neyman-Pearson, Bayes, etc.). We will denote by $D \in \mathcal{H} = \{H_0, H_1\}$ the output of the detector. The detection function given by (1) divides the subspace $\mathbb{R}^n$ in two disjoint regions, $\mathcal{R}$ and $\mathcal{R}^c$, termed *acceptance* or *detection region* and *rejection region*, respectively, such that $\mathbb{R}^n = \mathcal{R} \cup \mathcal{R}^c$, which are defined as

$$\mathcal{R} = \{\mathbf{z} \in \mathbb{R}^n : l(\mathbf{z}) > \eta\}; \mathcal{R}^c = \{\mathbf{z} \in \mathbb{R}^n : l(\mathbf{z}) \leq \eta\}.$$

Unfortunately, an analytical derivation of the likelihood ratio test is not always feasible, so we will consider instead a more general family of detection functions. Thus, the test performed by the detector is

$$g(\mathbf{z}, \boldsymbol{\theta}) \begin{array}{c} H_1 \\ > \\ \leq \\ H_0 \end{array} \eta,$$

where $\boldsymbol{\theta}$ is the secret key used in the detection process. Be aware that the resulting detector will be optimal only when $g(\mathbf{z}, \boldsymbol{\theta})$ coincides with the likelihood ratio $l(\mathbf{z})$.

In the considered scenarios, the watermark detector is often made public, generally in the form of a tamper-proof black box which only provides binary outputs, in such a way that an observer can check whether $g(\mathbf{a}, \boldsymbol{\theta})$ is larger or smaller than η, but he can not know its actual value. This scenario gives raise to the so-called *oracle attacks*, where the attacker can use the detector outputs to some selected inputs in order to gain knowledge about secret information used in the detection process (for instance, the detection key). Intuitively speaking, the detector acts as an oracle, responding *yes* or *no* to the inputs provided by the attacker. The most popular oracle attack is the so-called *sensitivity attack*,

introduced for the first time in [3]. At the time this attack was proposed, *additive spread spectrum* methods [4] constituted the state of the art in digital watermarking, so this attack was suited to this particular scenario. For additive spread spectrum under the assumption of a Gaussian host, the likelihood ratio has a well-known closed-form solution, given by $l(\mathbf{z}) = \mathbf{z}^T\mathbf{w}$, so the optimal detector in this case must apply the following test:

$$\mathbf{z}^T\mathbf{w} \underset{H_0}{\overset{H_1}{\gtrless}} \eta. \tag{2}$$

Detectors that implement the test given by (2) are termed *linear correlator detectors*. Essentially, the sensitivity attack (specialized to the case of digital images) for this kind of detectors consists of the following steps [3]:

1. The algorithm starts from a watermarked image $\mathbf{y}$ of dimension n. The first step is the modification of $\mathbf{y}$ so as to obtain a new image $\mathbf{z}$ near the boundary of $\mathcal{R}$, which according to (2) is a hyperplane in an n-dimensional subspace, perpendicular to $\mathbf{w}$.
2. For the i-th pixel of $\mathbf{z}$, a random vector $\mathbf{t}^i = [0, \cdots, 0, t_i, 0, \cdots, 0]^T$ is added to $\mathbf{z}$ observing how the sign of t_i affects the outputs of the detector and, hence, gaining knowledge about the polarity of the watermark in each pixel. Since $\mathbf{z}$ is near the detection boundary, small changes are likely to toggle the detector response. This procedure is repeated for all $i = 1, \cdots, n$.
3. At the end of the previous step, by combining the results for all pixels, the attacker has a rough estimate $\hat{\mathbf{w}}$ of the watermark vector and, thus, of the detection boundary, which in the considered case is perpendicular to $\mathbf{w}$.

According to the classification introduced at the beginning of this section, the sensitivity attack clearly falls into the category of attacks to security, since the attacker is trying to disclose the boundary of the detection region (which is supposed to be secret to unauthorized users). Of course, once the attacker has estimated this boundary, he can use his knowledge to devise smart attacks against watermarked contents: for instance, once the estimate $\hat{\mathbf{w}}$ has been obtained, the attacker can generate an attacked image $\mathbf{z}$ with small distortion, capable of fooling the detector, just by subtracting a suitably scaled version of $\hat{\mathbf{w}}$. Before the sensitivity attack was proposed, it was believed that the complexity of an attack disclosing the watermark was $O(2^n)$ (by means of a *brute force* approach), but the proposed strategy showed that it would be feasible in a number of iterations which is linear with the dimensionality of the watermarked image, i.e., the complexity of the attack was reduced to $O(n)$. Hence, it is easy to realize that this attack represented a serious threat to any watermarking scheme with a public detector available, and it raised up the problem of security in watermarking.

This paper is concerned with a generalization of the sensitivity attack, providing a formulation that encompasses most known watermark detection scheme with parameterizable and differentiable (but unknown to the attacker) detection

boundaries; in fact, our approach is suitable even for attacking QIM schemes, whereas the sensitivity attacks that had been devised so far were only aimed against spread spectrum methods. The rest of the paper is organized as follows: Section 2 provides an overview of previous works dealing with the characterization of this attack and the countermeasures proposed to increase the security of a watermarking system where public detectors are available. In Section 3, our new formulation of the problem is presented, and its application to some examples is given in Section 4.

2 Previous work and improvements

The sensitivity attack for detectors based on linear correlation, i.e., those given by (2), was extensively studied in [5] and [6]. Starting from the formulation of the attack given in [4], which was explained in the Introduction, the work in [5] proposes a countermeasure based on the randomization of the detection boundary: the basic idea is to define a region around the points that satisfy $\mathbf{z}^T\mathbf{w} = \eta$ where the decision of the detector is made random, in order to reduce the sensitivity of the detector to small changes in its inputs. Thus, the detection function is modified as follows:

$$D = \begin{cases} H_1, & \text{if } \mathbf{z}^T\mathbf{w} > \eta_2 \\ H_0, & \text{if } \mathbf{z}^T\mathbf{w} < \eta_1 \\ H_1 \text{ with probability } p(\mathbf{z}^T\mathbf{w}), & \text{if } \eta_1 \leq \mathbf{z}^T\mathbf{w} \leq \eta_2 \end{cases}, \tag{3}$$

where the two new thresholds η_1 and η_2 must be close to η so as not to degrade significantly the performance of the detector, and $p(r)$ verifies $p(\eta_1) = 0$ and $p(\eta_2) = 1$. The internal behavior of the detector is such that its outputs are deterministic, i.e., the response of the detector is always the same for a fixed input signal $\mathbf{z}$, in order to avoid the estimation of $p(r)$ simply by feeding the same $\mathbf{z}$ to the detector repeatedly. Anyway, estimation of the watermark is still possible. Let $\mathbf{z}'$ be a vector such that $\eta_1 \leq (\mathbf{z}')^T\mathbf{w} \leq \eta_2$, and $\boldsymbol{\epsilon}$ a random vector. For sufficiently small $\epsilon_i, i = 1 \cdots n$, and $\mathbf{z} = \mathbf{z}' + \boldsymbol{\epsilon}$, we have that $p(\mathbf{z}^T\mathbf{w}) = p((\mathbf{z}')^T\mathbf{w} + \boldsymbol{\epsilon}^T\mathbf{w}) \approx p((\mathbf{z}')^T\mathbf{w})$, so after trying a sufficiently large number of different vectors $\boldsymbol{\epsilon}$, the value of $p((\mathbf{z}')^T\mathbf{w})$ can be estimated simply by counting the number of outcomes that yield $D = H_1$. Similarly, for $\mathbf{t}^i = [0, \cdots, 0, t_i, 0, \cdots, 0]^T$ and $\mathbf{z}^i = \mathbf{z}' + \mathbf{t}^i + \boldsymbol{\epsilon}$, we have $(\mathbf{z}^i)^T\mathbf{w} = (\mathbf{z}')^T\mathbf{w} + t_i w_i + \boldsymbol{\epsilon}^T\mathbf{w} \approx (\mathbf{z}')^T\mathbf{w} + t_i w_i = (\mathbf{z}')^T\mathbf{w} \pm t_i\delta$, where in the last equality we have assumed that $w_i \in \{\pm\delta\}$. By means of a first order approximation, and assuming that $p(r)$ is differentiable, we can write $p((\mathbf{z}^i)^T\mathbf{w}) \approx p((\mathbf{z}')^T\mathbf{w} \pm t_i\delta) \approx p((\mathbf{z}')^T\mathbf{w}) \pm t_i\delta p'((\mathbf{z}')^T\mathbf{w})$, where $p'(r) \triangleq \frac{\partial p(r)}{\partial r}$ is the derivative of $p(r)$. Again, using enough different vectors $\boldsymbol{\epsilon}$, an estimate of $p((\mathbf{z}^i)^T\mathbf{w})$ can be obtained. By comparing this estimate to the previous estimate of $p(\mathbf{y}^T\mathbf{w})$, the sign of w_i can be inferred (as long as $p(r)$ is a monotically increasing function). In [5], the information leakage about the watermark provided by the detector outputs is quantified in an information-theoretic sense, and the shape of the optimum function $p(r)$ for $\eta_1 \leq r \leq \eta_2$ that minimizes the information leakage is given. It is easy to see that this countermeasure complicates the

sensitivity attack, but its complexity still remains linear with the dimensionality of the images. In fact, a practical method for estimating the watermark in this framework was devised in [6]. The method basically consists of the following steps:

1. Starting from a valid watermarked image $\mathbf{y}$, an image $\mathbf{z}'$ which yields $\eta_1 \leq (\mathbf{z}')^T\mathbf{w} \leq \eta_2$ is constructed by iteratively degrading $\mathbf{y}$.
2. The image $\mathbf{z}'$ is perturbed by the addition of zero-mean random vectors $\mathbf{t}$ with $t_i = \{\pm\delta\}$. If $\mathbf{w}$ and $\mathbf{t}$ are positively correlated, the detector will return $D = H_1$ with higher probability, so $\mathbf{t}$ will be taken as an approximation of $\mathbf{w}$; otherwise, if $D = H_0$, then $-\mathbf{t}$ will be taken as an estimate of $\mathbf{w}$.
3. By averaging the estimates obtained in the previous step, an approximation of $\mathbf{w}$ is obtained.

Following this approach it is possible to obtain reliable estimates of $\mathbf{w}$ in a number of iterations which is a small multiple of n, as it was shown in [6].

Another approach for performing a successful sensitivity attack was presented in [7]. The method is able to estimate the boundary of the acceptance region by modeling the attack as a classical adaptive filtering problem: it is easy to realize that the linear detection function given in (2) for additive spread spectrum can be thought of in terms of filtering $\mathbf{z}$ with a filter $\tilde{\mathbf{w}}$ such that $\tilde{w}_i = w_{n+1-i}\ \forall\ i = 1, \cdots, n$; furthermore, the attacker knows that $\mathbf{z} * \tilde{\mathbf{w}} = g(\mathbf{z}, \boldsymbol{\theta})$, where $*$ denotes the convolution operator, so if he/she can access the values of $g(\mathbf{z}, \boldsymbol{\theta})$, then using this signal as reference he can manage to construct an estimate of $\tilde{\mathbf{w}}$. The authors propose in [7] the use of the Least Mean Squares (LMS) algorithm in order to iteratively construct these estimates. Let $\hat{\mathbf{w}}_k$ be the estimate of $\tilde{\mathbf{w}}$ in the k-th iteration and $\{\mathbf{z}_k\}$ a set of vectors near the detection boundary; each iteration of the LMS algorithm consists of the following steps:

$$
\begin{aligned}
&1. \quad r_k = \mathbf{z}_k * \tilde{\mathbf{w}},\\
&2. \quad e_k = g(\mathbf{z}_k, \boldsymbol{\theta}) - r_k,\\
&3. \quad \mathbf{w}_{k+1} = \mathbf{w}_k + \mu e_k \mathbf{z}_k,
\end{aligned}
$$

where μ is the step-length. In a more realistic situation, the attacker only has access to the detector outputs, D, so this algorithm must be properly modified. In this situation, the attacker must restrict the set $\{\mathbf{z}_k\}$ to those vectors lying near the detection boundary, because he still knows that $g(\mathbf{z}_k, \boldsymbol{\theta}) \approx \eta$; thus, the algorithm is complicated by the fact of computing the appropriate set $\{\mathbf{z}_k\}$. The authors also propose some modifications in order to cope with the countermeasure introduced in [5], which was explained above.

In view of the security flaws presented by traditional spread spectrum methods under sensitivity-like attacks, researchers put their effort in the design of *asymmetric* schemes [8].[2] One of the advantages offered by asymmetric schemes

[2] Watermarking techniques can be roughly classified according to the role of the secret key in the embedding/detection processes: those methods which use different keys for embedding and detection are termed *asymmetric*, otherwise they belong to the category of *symmetric schemes*.

against sensitivity attacks is the fact that the embedding and detection keys are different, thus the impact of a successful attack revealing the detection boundary is minimized (recall that disclosure of the watermark in traditional spread spectrum methods allows to unwatermark legal contents, as well as generating forged illegal documents). The other advantage of asymmetric watermarking is the use of more involved detection regions, complicating the description of the detection boundary; for instance, in [8], four asymmetric methods are analyzed under a unified framework, showing that the detection function can be written in terms of a quadratic form in $\mathbb{R}^n$ for all cases, i.e.

$$\frac{\mathbf{z}^T\mathbf{A}\mathbf{z}}{n} \underset{H_0}{\overset{H_1}{\gtrless}} \eta.$$

The idea of increasing the security of the system against sensitivity attacks by complicating the detection region is exploited by the family of detection functions called JANIS [9], which use N-th order polynomial detection functions, i.e.

$$g(\mathbf{z}, \boldsymbol{\theta}) = \frac{1}{n} \sum_{k=1}^{n/N} \prod_{j=1}^{N} z_{p[(k-1)\cdot N+j]} \cdot a_{p[(k-1)\cdot N+j]},$$

where $\mathbf{a}$ is secret random ± 1 vector and $\mathbf{p}$ is a secret random permutation vector. Based on this detection function the watermark is obtained as $\mathbf{w} = \gamma \nabla g(\mathbf{x}, \boldsymbol{\theta})$, where γ is a parameter to adjust the embedding distortion. Indeed it makes more difficult the sensitivity attack, but obviously this is not the ultimate solution: for example, a N-th order detection boundary can still be described by estimating n^N points on the detection boundary. This point was addressed in [7], showing that the LMS attack can be properly modified in order to cope with this kind of detection boundaries. A possible solution to this problem was proposed also in [7] by means of non-parametric decision boundaries, i.e., by using decision boundaries that can not be described by a finite number of parameters. An example of such decision boundaries are those given by fractal curves like the Peano curve, which is used in [7] to replace the original linear detection boundary in a spread spectrum scheme. With a proper design, the proposed method can invalidate sensitivity attacks with slight degradations in robustness.

Recently, an attempt to give a rigorous formulation of the sensitivity attack was presented in [10]: first, the convergence of the algorithm proposed in [6] is proven, using the law of large numbers; thereafter, a new non-iterative sensitivity attack for detectors based on linear correlation is presented.[3] The main steps of this new algorithm are outlined in the following:

1. As in the former algorithms, the first step is the construction of a signal $\mathbf{z}'$ near the boundary of the detection region.

[3] As a further contribution, this new algorithm is also suitable for estimating continuous-valued watermarks, whereas the algorithms previously proposed in [5] and [6] assumed that the watermark could only take discrete values.

2. Now consider the set of vectors $\{\mathbf{t}^i\}, \quad i = 1, \cdots, n$, defined by the canonical basis of $\mathbb{R}^n$. For each $\mathbf{t}^i$, a signal $\mathbf{z}'' = \mathbf{z}' + \alpha_i \mathbf{t}^i$ on the detection boundary is constructed, by properly selecting the scaling factor α_i. The search for this value of α_i must be accomplished by means of some numerical algorithm, so it will be surely the most costly part of the algorithm.
3. For the detector under consideration, it holds that $(\mathbf{z}'')^T\mathbf{w} = \sum_{k=1}^{n} z''_k w_k + \alpha_i w_i = \eta, i = 1, \cdots, n$ where η is the detection threshold and $w_i = (\mathbf{t}^i)^T\mathbf{w}$. Thus, a linear system with n equations and n unknowns has been defined. By taking into account the special structure of this system, it is easy to show that it can be solved in $n + 1$ elemental operations.

Another remarkable contribution of [10] is the extension of the sensitivity attack in order to work with a more generic family of detection functions of the form $g(\mathbf{y}, \mathbf{w})$; furthermore, this method has the advantage of return an estimate of the watermark. Nevertheless, this approach presents several drawbacks: the attacker needs to know the detection function and even the inverse of the gradient of the detection function. Thus, the need for a new formulation which overcomes these problems is justified; in the next section we will try to solve this problem, achieving a solution which will be shown to work with a wider range of detection functions. The method proposed has the following characteristics:

- It does not require knowledge about the detection function; it just needs to know the binary output of the detection function for a given input. Due to this, our method is indeed able to deal with watermarking methods which use a secret detection key (different from the embedding key), in such a way that the attacker has no access to the decoding function; these methods are known under the generic name of *asymmetric watermarking* (see [8] and [11]).
- The gradient of the detection function does not need to be inverted. As it was said in the previous point, sometimes the detection function will not be known by the attacker, so he/she will not be able to invert its gradient.

3 The Blind Newton Sensitivity Attack (BNSA)

Focusing on watermark detection, we will describe the detector output through the function $f_{\text{binary}} : \mathbb{R}^n \to \mathcal{H}$, with $\mathcal{H} = \{H_0, H_1\}$. Without loss of generality, we can define the following functions

$$\begin{aligned} f &: \mathbb{R}^n \to \mathbb{R}^m \text{ and} \\ g_{\text{binary}} &: \mathbb{R}^m \to \mathcal{H}, \end{aligned} \tag{4}$$

with $m \leq n$, in such a way that $f_{\text{binary}} = g_{\text{binary}} \circ f$, and f is parameterized by the secret key $\boldsymbol{\theta}$. This decomposition will be shown to be useful in the next sections, since some of the most popular watermarking techniques perform embedding/detection in a projected domain so f can be seen as the projection function. Furthermore, in the schemes studied in this paper the output of g_{binary}

will be based on the output of a real function g and a threshold η, in such a way that

$$g_{\text{binary}}(\mathbf{x}) = \begin{cases} H_0, \text{ if } g(\mathbf{x}) \leq \eta \\ H_1, \text{ if } g(\mathbf{x}) > \eta \end{cases}, \tag{5}$$

with $g : \mathbb{R}^m \to \mathbb{R}$.

On the other hand, a distorsion measure has to be defined in order to quantify the impact of the attacking signal $\mathbf{t}$ on the watermarked signal $\mathbf{y}$: [4]

$$\begin{aligned} d_{\mathbf{y}} : \mathbb{R}^n &\to \mathbb{R}^+ \\ \mathbf{t} &\to d_{\mathbf{y}}(\mathbf{t}). \end{aligned}$$

This distortion measure could be based on perceptual criteria (depending on the nature of the host signal), although very often, and due to simplicity, the squared Euclidean norm of $\mathbf{t}$ is chosen (i.e., $d_{\mathbf{y}}(\mathbf{t}) = ||\mathbf{t}||_2^2$).

Recalling that the attacker tries to find the vector $\mathbf{t}$ which yields a "no watermark" decision (i.e., $f_{\text{binary}}(\mathbf{y} + \mathbf{t}) = H_0$) while minimizing the distortion measure $d_{\mathbf{y}}(\mathbf{t})$, his/her target can be formalized as

$$\arg \min_{\mathbf{t}: g \circ f(\mathbf{y}+\mathbf{t}) \leq \eta} d_{\mathbf{y}}(\mathbf{t}). \tag{6}$$

Let us assume that $d_{\mathbf{y}}(\mathbf{t})$ is a continuous and convex function of $\mathbf{t}$ (for a given watermarked signal $\mathbf{y}$), which achieves its absolute minimum value at $\mathbf{t}_0$ (the squared Euclidean norm obviously fulfills these conditions), a vector that belongs the set of attacking vectors yielding H_1 (which we will denote by $\mathcal{B}$),[5] i.e., $\mathbf{t}_0 \in \mathcal{B} \triangleq \{\mathbf{t} : g \circ f(\mathbf{t} + \mathbf{y}) > \eta\}$. Then, replacing $\mathcal{B}$ in (6), and denoting by $\partial\mathcal{B}$ its boundary and by $\mathcal{B}^c$ its complement, it is straightforward to show that$\arg\min_{\mathbf{t} \in \mathcal{B}^c} d_{\mathbf{y}}(\mathbf{t}) \in \partial\mathcal{B}$, so (6) is tantamount to

$$\arg \min_{\mathbf{t}: g \circ f(\mathbf{y}+\mathbf{t}) = \eta} d_{\mathbf{y}}(\mathbf{t}). \tag{7}$$

This is a typical Lagrange's multipliers problem, so the attacker could find a theoretical solution if both d and $g \circ f$ were known by him/her; nevertheless, this is not the case, since the last one depends on the secret key, which is unknown for the attacker. Actually, he/she will have only access to the binary output of the decoder. In Appendix A we will show that this is equivalent to

$$\arg \min_{\mathbf{s} \in \mathbb{R}^n} d^{\star}_{\mathbf{y}}(h_{\mathbf{y}}(\mathbf{s})), \tag{8}$$

[4] Ideally this measure should quantify the differences between the original host signal and its attacked version; nevertheless, the attacker will have to design his/her strategy taking into account the watermarked signal, since he/she has not access to the original one.

[5] Be aware that in most cases it is reasonable to consider that $\mathbf{t}_0 = \mathbf{0}$, since in that case the attacked signal will be the watermarked one, so the distortion is minimized; furthermore $\mathbf{t}_0$ is in $\mathcal{B}$, since $g \circ f(\mathbf{y})$ will yield H_1.

where $d^\star_{\mathbf{y}}$ the restriction of $d_{\mathbf{y}}$ to the boundary of $\mathcal{B}$, and $h_{\mathbf{y}}$ is a surjection which maps $\mathbb{R}^n$ onto the boundary of the decision region.

Since theoretical solutions to (8) are not in general possible due to the lack of knowledge of the boundary of the decision region, numerical iterative methods should be applied (in general) by the attacker in order to find the solution. Concretely, in this paper we will use an adaptation of Newton's method [12], where the considered vector in the $(k+1)$-th iteration is computed as

$$\mathbf{s}_{k+1} = \mathbf{s}_k - \xi_k \cdot \left[\nabla^2 (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)\right]^{-1} \cdot \nabla (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k), \tag{9}$$

where $\xi_k \in \mathbb{R}^+$ is the step-length, whose computation requires (in general) a line search [12]: a small value of ξ_k will imply a slow convergence, but with a large one convergence cannot be assured. When the boundary to be estimated is known to be an hyperplane we can adopt $\xi_k = 1$.

It is straightforward to see that $\nabla (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)$ and $\nabla^2 (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)$ cannot be obtained in an analytic way, therefore they must be numerically approximated by taking into account that

$$\begin{aligned}
\frac{\partial (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})}{\partial s_i}(\mathbf{s}) &= \frac{(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s} + \delta \mathbf{e}_i) - (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})}{\delta} + O(\delta), \text{ and} \\
\frac{\partial^2 (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})}{\partial s_i \partial s_j}(\mathbf{s}) &= \frac{(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s} + \delta \mathbf{e}_i + \delta \mathbf{e}_j) - (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s} + \delta \mathbf{e}_i)}{\delta^2} \\
&\quad + \frac{-(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s} + \delta \mathbf{e}_j) + (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})}{\delta^2} + O(\delta),
\end{aligned}$$

with $\mathbf{e}_i$ the i-th vector of the canonical basis. Another choice, which is especially suitable for large-scale problems, is based on replacing the Hessian by a diagonal matrix keeping the diagonal elements; in that way, an iteration of the algorithm just requires $(2 \cdot n + 1)$ evaluations of $(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})$ and (9) is computed with n scalar divisions (if the complete matrix were used, a linear system with n equations and n variables should be solved).

On the other hand, $h_{\mathbf{y}}(\mathbf{s})$ is usually based on scaling $\mathbf{s}$ by a factor $\alpha \in \mathbb{R}$, such that $\alpha \cdot \mathbf{s} \in \partial\mathcal{B}$. The existence of such an α is based on the fact that for most of the known detection functions $\mathbf{0} \in \mathcal{B}$ and $\beta \cdot \mathbf{s} \in \mathcal{B}^c$ for large values of β, so α can be found by a dicothomy algorithm. Be aware that this method is based on the binary output of the detector, without any other knowledge about the detection function; this is why the algorithm is said to be *blind*.

4 Application to real methods

In this section we will particularize the proposed algorithm to some of the most popular watermarking methods, showing the practical usefulness of this new attack and comparing the performance of the different methods. In order to

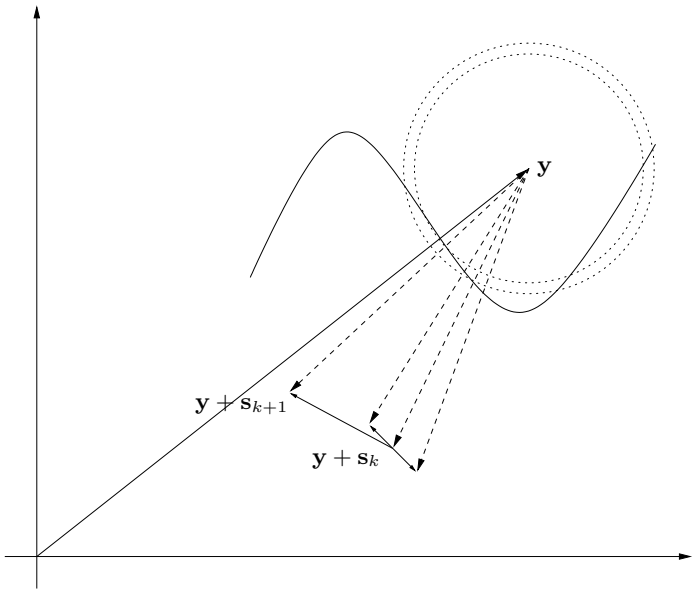


Fig. 1. Example of an iteration of the algorithm. Given a watermarked signal $\mathbf{y}$ and the attacking vector in the k-th iteration $\mathbf{s}_k$, the last one is slightly modified to estimate the gradient and Hessian of $d^\star_{\mathbf{y}} \circ h_{\mathbf{y}}(\mathbf{s}_k)$. Once the descent direction and the step-length have been computed, $\mathbf{s}_{k+1}$ is obtained. It can be seen that $\mathbf{y} + \alpha_{k+1}\mathbf{s}_{k+1}$ is closer to the boundary than $\mathbf{y} + \alpha_k \mathbf{s}_k$.

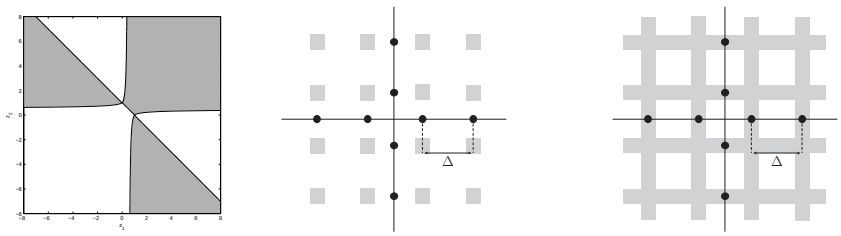
make a fair comparison, the value of the probability of false alarm P_{fa}[6] will be fixed to 10^{-4}, $n = 2048$ and the document to watermark ratio to 16 dB (with $\sigma^2_W = 1$) in order to ensure a reasonable probability of missed detection for all the studied methods.[6]

4.1 Spread Spectrum

Detection of standard Spread Spectrum methods is based on the correlation between the received signal $\mathbf{z}$ and the watermark $\mathbf{w}$. Therefore, the function f, defined in (4), projects $\mathbf{z}$ onto a one-dimensional domain ($m = 1$), i.e. $f(\mathbf{z}) = \mathbf{z}^T \cdot \mathbf{w}$, and g in (5) will be the identity function ($g(x) = x$, for all $x \in \mathbb{R}$), so the detection is given by

$$\mathbf{z}^T \cdot \mathbf{w} \underset{H_0}{\overset{H_1}{\gtrless}} \eta,$$

[6] The probability of false alarm P_{fa} is defined as $\Pr\{g_{\text{binary}} \circ f(\mathbf{x} + \mathbf{t}) = H_1\}$. On the other hand, the probability of missed detection P_m is defined as $\Pr\{g_{\text{binary}} \circ f(\mathbf{x} + \mathbf{w} + \mathbf{t}) = H_0\}$.



(a) (b) (c)

Fig. 2. Examples of decision regions: (a) Decision regions obtained taking into account a l_c-norm when $c = 0.5$. (b) AND Region for QPD. (c) OR Region for QPD.

in such a way that the boundary of the decision region ($\partial\mathcal{B}$) will be a hyperplane and just one iteration will be needed to estimate its orthogonal vector, i.e. the projecting vector, which is a scaled version of the watermark itself. Nevertheless, due to the approximation of the Hessian by a diagonal matrix, about 4 iterations are needed to meet a tolerance of 10^{-6} in the squared norm of the optimal attacking vector. As it was said in Section 3, the line search is not necessary in this case and ξ_k can be fixed to 1.

Comparing the cost of this method with that proposed by El Choubassi and Moulin [10], the latter requires the knowledge of only n points in the border to estimate the watermark, whereas we need $8 \cdot n$ points.

Another alternative for the detection function is that proposed by Cox *et al.* in [13]; in that case, f quantifies the angle between the received signal $\mathbf{z}$ and the watermark vector $\mathbf{w}$, i.e. $f(\mathbf{z}) = \frac{\mathbf{z}^T \cdot \mathbf{w}}{||\mathbf{z}|| \cdot ||\mathbf{w}||}$, and g is again the identity function, yielding a decision region $\mathcal{B}$ which is a n-dimensional cone.

4.2 Side-informed methods

In Section 2 the JANIS methods were introduced. In order to make a comparison with the other existing methods, we have fixed the order of the detection function to 4, so

$$f(\mathbf{z}) = \frac{1}{n} \sum_{k=1}^{n/4} \prod_{j=1}^{4} z_{p[(k-1)\cdot N+j]} \cdot a_{p[(k-1)\cdot N+j]}.$$

Quantization-based methods have been shown to be useful for data hiding applications; nevertheless, and despite of their success in that application, very little has been said about their use in detection scenarios. To the best of our knowledge, the first work addressing the problem from this point of view was [14], where the Scalar Costa Scheme is adapted to authentication purposes by embedding a fixed message, yielding the detection function $g(\mathbf{z}, \boldsymbol{\theta}) = \frac{f_{\mathbf{Y}}(\mathbf{z})}{f_{\mathbf{X}}(\mathbf{z})}$. Note that in this case the sensitivity attack is straightforward, since it can be done componentwise.

On the other hand, in [15] the received signal $\mathbf{z}$ is quantized with a lattice Λ and the decision is made upon the squared norm of the quantization error. Formalizing it, we can write $f(\mathbf{z}) = ||\mathbf{z} \mod \Lambda||^2$, and g is the identity function again. In this way, the acceptance region is the union of n-dimensional hyperspheres centered at the centroids of Λ. From the point of view of attacking such a system, this decision region assures that the attacker can produce a signal yielding H_0 by adding *any* noise vector with a given variance, as far as that noise vector is independent of the self noise. Therefore, a sensitivity attack is not really necessary in this case.

Another approach to this problem is Quantized Projection based Detection (QPD) [16], where uniform scalar quantizers are used to quantify a m-dimensional projected version of the received signal $\mathbf{z}$ and the detection function depends on the quantization error, introducing two different strategies: the AND and OR detection regions, which can be formalized as

$$\begin{aligned} f_i(\mathbf{z}) &= \sum_{j=1}^{n} a_{ij} z_j,\ 1 \le i \le m, \\ g_{\text{AND}}(f(\mathbf{z})) &= \max_{1 \le i \le m} |(f_i(\mathbf{z}) \mod \Delta) - \Delta/2|, \text{and} \\ g_{\text{OR}}(f(\mathbf{z})) &= \min_{1 \le i \le m} |(f_i(\mathbf{z}) \mod \Delta) - \Delta/2|, \end{aligned}$$

where Δ is the quantization step, a_{ij} are the secret projection matrix coefficients and m the dimensionality of the projected subspace. Obviously, the optimal attacking strategy will depend on the chosen decision region. The convergence of the algorithm introduced in Section 3 for finding the optimal attacking vector will be very much slower for the OR region, since the cost function has its minimum value at a non-differentiable point. In fact, in such case we will follow a different strategy in which we try to estimate the m projecting vectors to compute the optimal attacking vector as the sum of them, which implies the complete disclosure of the secret key.

4.3 Comparison

In Fig. 4.3 the power needed to achieve an unwatermarked signal is plotted versus the number of iterations of BNSA; we can see that the power needed at iteration 0 (just randomly generated vectors) is much larger for SS based on an hyperplane, but converges to that of angle-based SS when the number of iterations is increased. In the same way, the most robust method against the BNSA among those plotted in the figure is JANIS, even when the power required for producing an unwatermarked signal is reduced in 24 dB after 10 iterations. For QPD-AND, as soon as one of the projecting vectors has been estimated, the power needed to yield an unwatermarked signal is significantly smaller than in the other studied cases. Finally, for QPD-OR, the power required after 10 iterations is only -38 dB.

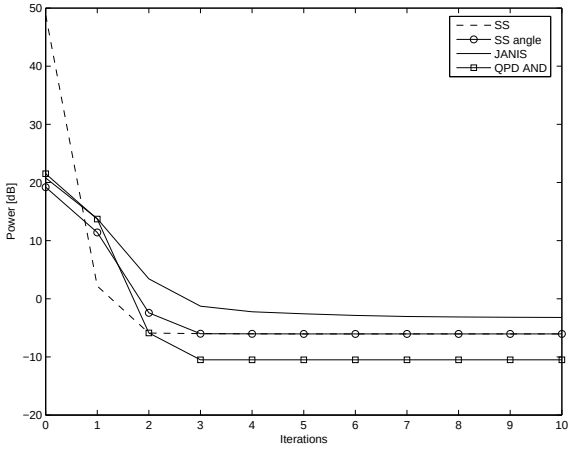


Fig. 3. Power needed to yield an unwatermarked signal (in dB) averaged over 100 watermarked Gaussian vectors as a function of the number of iterations (0 when there is not attack), for different decision regions: SS based on a hyperplane, SS based on the angle, JANIS and QPD for AND regions. Iteration 0 corresponds to random attacking vectors (without applying the proposed algorithm).

5 Final remarks

Following are some guidelines on how to measure the robustness of watermarking methods against BNSA, the design of practical watermarking methods which are BNSA-resistant, and the application of BNSA to new scenarios:

- The power needed to push a watermarked signal out of the detection region after the BNSA can be seen as a measure of the robustness of a watermarking method against this attack: the larger the power needed, the more robust the method is. In this sense, JANIS could be said to be the most robust among the studied methods, whereas the QPD methods show quite poor performance. Note, however, that this measure does not provide full information on the behavior of a particular method; for instance, QPD methods, which been shown here to be quite weak against BNSA, have a very good Receiver Operating Characteristic (see [16] for a comparison with SS).
- As a countermeasure against BNSA, one could design detection functions for which component-wise modifications produce bounded increments, since for this kind of functions the task of finding vectors on the boundary of the detection function is considerably complicated. Interestingly, the ML detection function for Generalized Gaussian distributed hosts (which is a l_c-norm, see [17]), fulfills this requirement whenever the shape parameter c is such that $c < 1$.
- Taking into account that it just needs the binary output of the detector, the BNSA is also suitable for zero-knowledge protocols [18], where, at the end,

regardless of the domain where the detection function is computed, there is a detection region which can be estimated by the proposed algorithm.
- As a final remark, the approach presented in this paper can be also used in the case of data-hiding systems, since the decoding process is nothing but a multiple hypothesis test. In this case, any change of the decoder output should be interpreted as if it were done by a change in the detector output; this is equivalent to have the following binary hypothesis: a) the decoded message is changed; b) the decoded message is unaltered.

A Appendix

In this Appendix we will show that (7) is equivalent to

$$\arg \min_{\mathbf{s} \in \mathbb{R}^n} d^{\star}_{\mathbf{y}}(h_{\mathbf{y}}(\mathbf{s})), \tag{10}$$

with $d^{\star}_{\mathbf{y}}(\mathbf{t})$ the restriction of $d_{\mathbf{y}}(\mathbf{t})$ to those $\mathbf{t} \in \partial\mathcal{B}$, i.e.,

$$\begin{aligned} d^{\star}_{\mathbf{y}}(\mathbf{t}) : \partial\mathcal{B} &\rightarrow \mathbb{R}^{+} \\ \mathbf{t} &\rightarrow d_{\mathbf{y}}(\mathbf{t}), \end{aligned}$$

and $h_{\mathbf{y}}(\mathbf{s})$ is a surjection from $\mathbb{R}^n$ to $\partial\mathcal{B}$, i.e.,[7] $h_{\mathbf{y}}(\mathbf{s}) : \mathbb{R}^n rightarrow \partial\mathcal{B}$, such that $h_{\mathbf{y}}(\mathbb{R}^n) = \partial\mathcal{B}$, verifying that $h_{\mathbf{y}}(\mathbf{s}) = \mathbf{s}$ for all $\mathbf{s} \in \partial\mathcal{B}$; we will also assume that $h_{\mathbf{y}}(\mathbf{s}) \in C^2$, i.e., its second derivative exists and is continuous, in a neighborhood of $\mathbf{s}$ (this last point is related to the differentiability of $g \circ f$). Note that $h_{\mathbf{y}}(\mathbf{s})$ just maps the vector $\mathbf{s}$ to a point on $\partial\mathcal{B}$; following this approach the constraint in (7) is straightforwardly verified and we no longer have to care about it. In this way, if $\mathbf{t}_1^*$ is a solution to (7), it will verify $g \circ f(\mathbf{y} + \mathbf{t}_1^*) = \eta$, so $\mathbf{t}_1^* \in \partial\mathcal{B}$ and we can define the set of vectors $\mathcal{S}_1 \triangleq \{\mathbf{s}_1^* \in \mathbb{R}^n : h_{\mathbf{y}}(\mathbf{s}_1^*) = \mathbf{t}_1^*\}$. Taking into account that $h_{\mathbf{y}}$ is a surjection there will be at least one such vector $\mathbf{s}_1^* \in \mathcal{S}_1$, so $d^{\star}_{\mathbf{y}}(h_{\mathbf{y}}(\mathbf{s}_1^*)) = d_{\mathbf{y}}(\mathbf{t}_1^*)$, and $\mathbf{s}_1^*$ is a solution to (10). On the other hand, if $\mathbf{s}_2^*$ is a solution to (10), we can define $\mathbf{t}_2^* = h_{\mathbf{y}}(s_2^*)$, which minimizes $d^{\star}_{\mathbf{y}}(\mathbf{t})$ over $\partial\mathcal{B}$, so $\mathbf{t}_2^*$ also minimizes $d\mathbf{y}(\mathbf{t})$ for all $\mathbf{t} \in \partial\mathcal{B}$, and is a solution to (7).

Therefore, a vector $\mathbf{s}$ is a solution to (10) if and only if $h_{\mathbf{y}}(\mathbf{s})$ is a solution to (7), in such a way that we can restrict our problem to look for a function $h_{\mathbf{y}}$ and an algorithm which finds a solution to (10).

References

1. Kalker, T.: Considerations on watermarking security. In: IEEE International Workshop on Multimedia Signal Processing, MMSP'01, Cannes, France (2001) 201–206
2. Comesaña, P., Pérez-Freire, L., Pérez-González, F.: Fundamentals of data hiding security and their application to Spread-Spectrum analysis. In: 7th Information Hiding Workshop, IH05. Lecture Notes in Computer Science, Barcelona, Spain, Springer Verlag (2005)

[7] This means that for all $\mathbf{b} \in \partial\mathcal{B}$, there is an $\mathbf{a} \in \mathbb{R}^n$ such that $h_{\mathbf{y}}(\mathbf{a}) = \mathbf{b}$.

3. Cox, I.J., Linnartz, J.P.M.G.: Public watermarks and resistance to tampering. In: IEEE International Conference on Image Processing ICIP'97. Volume 3., Santa Barbara, California, USA (1997) 3–6
4. Cox, I.J., Killian, J., Leighton, T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing **6** (1997) 1673–1687
5. Linnartz, J.P.M.G., van Dijk, M.: Analysis of the sensitivity attack against electronic watermarks in images. In Aucsmith, D., ed.: 2nd International Workshop on Information Hiding, IH'98. Volume 1525 of Lecture Notes in Computer Science., Portland, OR, USA, Springer Verlag (1998) 258–272
6. Kalker, T., Linnartz, J.P., van Dijk, M.: Watermark estimation through detector analysis. In: IEEE International Conference on Image Processing, ICIP'98, Chicago, IL, USA (1998) 425–429
7. Mansour, M.F., Tewfik, A.H.: LMS-based attack on watermark public detectors. In: IEEE International Conference on Image Processing, ICIP'02. Volume 3. (2002) 649–652
8. Furon, T., Venturini, I., Duhamel, P.: An unified approach of asymmetric watermarking schemes. In Edward J. Delp III, Wong, P.W., eds.: Security and Watermarking of Multimedia Contents III. Volume 4314., San Jose, California, USA, SPIE (2001) 269–279
9. Furon, T., Macq, B., Hurley, N., Silvestre, G.: JANIS: Just Another N-order side-Informed watermarking Scheme. In: IEEE International Conference on Image Processing, ICIP'02. Volume 3., Rochester, NY, USA (2002) 153–156
10. El Choubassi, M., Moulin, P.: New sensitivity analysis attack. In Edward J. Delp III, Wong, P.W., eds.: Security, Steganography and Watermarking of Multimedia contents VII, SPIE (2005) 734–745
11. Furon, T., Duhamel, P.: An asymmetric watermarking method. IEEE Trans. on Signal Processing **51** (2003) 981–995 Special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery.
12. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer (1999)
13. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital watermarking. Multimedia Information and Systems. Morgan Kauffman (2002)
14. Eggers, J.J., Girod, B.: Blind watermarking applied to image authentication. In: Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). Volume 3. (2001) 1977–1980
15. Liu, T., Moulin, P.: Error exponents for one-bit watermarking. In: Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). Volume 3. (2003) 65–68
16. Pérez-Freire, L., Comesaña, P., Pérez-González, F.: Detection in quantization-based watermarking: performance and security issues. In Edward J. Delp III, Wong, P.W., eds.: Security, Steganography, and Watermarking of Multimedia Contents VII. Volume 5681., San Jose, California, USA, SPIE (2005) 721–733
17. Hernández, J.R., Amado, M., Pérez-González, F.: DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure. IEEE Trans. on Image Processing **9** (2000) 55–68 Special Issue on Image and Video Processing for Digital Libraries.
18. Adelsbach, A., Sadeghi, A.R.: Zero-knowledge watermark detection and proof of ownership. In Moskowitz, I.S., ed.: 4th International Workshop on Information Hiding, IH'01. Volume 2137 of Lecture Notes in Computer Science., Pittsburgh, PA, USA, Springer Verlag (2001) 273–288