Watermarking security

Teddy Furon

Chapter 6

Watermarking security

Teddy Furon

This chapter deals with applications where watermarking is a security primitive included in a larger system protecting the value of multimedia content. In this context, there might exist dishonest users, in the sequel so-called attackers, willing to read/overwrite hidden messages or simply to remove the watermark signal.

The goal of this section is to play the role of the attacker. We analyze means to deduce information about the watermarking technique that will later ease the forgery of attacked copies. This chapter first proposes a topology of the threats in Section 6.1, introducing three different concepts: robustness, worst-case attacks, and security. Chapter **??** has already discussed watermark robustness. We focus on worst-case attacks in Section 6.2, on the way to measure watermarking security in Section 6.3, and on the classical tools to break a watermarking scheme in Section 6.4. This tour of watermarking security concludes by a summary of what we know and still do not know about it (Section 6.5) and a review of oracle attacks (Section 6.6). Last, Section 6.7 deals with protocol attacks, a notion which underlines the illusion of security that a watermarking primitive might bring when not properly used in some applications.

6.1 INTRODUCTION

Chapter **??** already mentioned the term "attacks." This introduction defines a topology to clearly underpin how different the "attacks" of this chapter are. This

topology has three classes: robustness attacks, worst-case attacks, and security attacks. They range from the case where the attacker knows nothing about the watermarking technique (robustness) to the case where the attacker is willing to disclose all its internals and especially its secret key (security).

6.1.1 Robustness

The previous chapter calls an attack any process modifying multimedia content which may decrease the performance of the watermark detection or decoding. This acts as the communication channel that the watermark signal goes through. Many such processes are expected in the lifetime of multimedia content. They are routinely used during editing or rendering without the intention to hurt the watermark. They are not designed for watermarking removal but for source compression, denoising, filtering, etc. These attacks are *blind* in the sense that one uses them not having the slightest idea about what watermarking is. Section **??** classified them into synchronous (or valuemetric) attacks that change the value of the samples and the asynchronous (or geometric) attacks. The localization of the watermark signal in the spatial/time and/or frequency domain has moved so that the decoder will not look for it at the right place.

State-of-the-art watermarking techniques are very robust against synchronous or asynchronous attacks; being robust to a combination of both is more challenging. In details, watermarking modifies the samples in the embedding domain as follows:

$$\mathbf{c}_{\mathrm{w}} = \mathbf{c}_{\mathrm{o}} + \mathbf{w}(\mathbf{c}_{\mathrm{o}}, \mathbf{s}, \mathbf{m}, K) \tag{6.1}$$

where $\mathbf{c}_{\mathrm{o}} = (\mathbf{c}_{\mathrm{o}}[1], \ldots, \mathbf{c}_{\mathrm{o}}[N])$ and $\mathbf{c}_{\mathrm{w}} = (\mathbf{c}_{\mathrm{w}}[1], \ldots, \mathbf{c}_{\mathrm{w}}[N])$ are the host and watermarked samples, respectively, extracted from the original and the watermarked pieces of content. Function $\mathbf{w}(\cdot)$ is the embedding scheme that creates the watermark signal from the host $\mathbf{c}_{\mathrm{o}}$, the perceptual masking slack $\mathbf{s}$, the message to be hidden $\mathbf{m}$, and the secret key K. The embedding distortion is denoted D_e. A synchronous attack produces some noise in the embedding domain:

$$\mathbf{c}_{\mathrm{z}} = \mathbf{c}_{\mathrm{w}} + \mathbf{n} \tag{6.2}$$

We can quantify its impact by measuring how the performance P of the watermarking scheme decreases as the amount of distortion D_a between the watermarked and the attacked contents increases.

- P is often measured in practice by the bit error rate (BER) or the message error rate (MER) in watermark decoding, or the probability of a miss P_{miss}

in watermark detection. In theoretical papers, it is expressed in terms of mutual information or capacity for watermark decoding or Kullback-Leibler or Bhattacharyya distances for watermarking detection.

- D_a is often the mean square error (MSE) between watermarked and attacked (or watermarked and original) contents, be it translated into PSNR in still image watermarking, or SNR in audio watermarking, or the expectation of the Euclidean distance in theoretical papers.

As is often the case, the attack is parametrized by a setup θ_a, giving birth to the operating curve $(P(\theta_a), D_a(\theta_a))$. For example, as the quality factor θ_a of a JPEG compression goes down, the distortion $D_a(\theta_a)$ between the compressed and the watermarked image gets bigger, while the performance $P(\theta_a)$ of the watermarking decoder or detector smoothly decreases.

6.1.2 Worst-Case Attacks

The concept of worst case attacks is summarized into one question: for a given attack distortion D_a, which process minimizes the watermarking performance P?

This concept is of utmost importance, as it yields a fair benchmark. Imagine that watermarking technique A is more robust to JPEG compression than technique B, but the latter is more robust to JPEG2000. Which technique is more robust? Not considering a particular attack, but on the contrary, focusing on the one that hurts most, reveals the intrinsic robustness of a particular watermarking scheme. Techniques A and B do not have the same worst-case attack, but their ultimate robustness allows to predict how they will resist against an *informed* attacker. This last word is the keystone of the concept. The nature of the worst-case attack mostly depends on the information available to the attacker.

The worst-case attack should be defined as the most damaging attack at a given distortion budget and for a given level of knowledge on the watermarking technique.

The literature always looks for the worst-case attack, assuming that the attacker knows the embedding domain and the watermarking scheme. In other words, he has access to the samples that carry the watermarking signal (6.1). Then, the attacker looks for the noise signal $\mathbf{n}$ to be added (6.2) to degrade, at most, the system performance. It is therefore tweaked for a particular watermarking embedding $\mathbf{w}(\cdot)$. Yet, the attacker does not know secret key K and cannot set $\mathbf{n} = -\mathbf{w}(\mathbf{c}_o, \mathbf{s}, \mathbf{m}, K)$. Signal $\mathbf{n}$ is thus a random noise that produces a given performance P for an expected attack distortion D_a. The goal is to find the

distribution of random noise that maximizes the loss of performance at a given distortion budget.

Examples of worst-case attacks against two well known watermarking schemes (spread spectrum and quantization index modulation) are detailed in Section 6.2.

6.1.3 Security Attacks

6.1.3.1 Differences Between Robustness and Security

The concept of watermarking security has slowly emerged because it was often misunderstood as a synonym of robustness; this is because both terms deal with attacks. The intention of removing the watermark or the malice of the attacker are not sufficient enough to make a clear cut between the two concepts.

During security attacks, the attacker proceeds in two steps: he first observes some protected contents in order to disclose some information about the watermarking technique and its secret key. He then forges attacked contents thanks to a worst-case attack based on this new knowledge.

Security becomes a concern when the attacker observes many pieces of content watermarked with the same technique and the same secret key.

6.1.3.2 An Analysis Based on Attacker's Knowledge

Following Kerckhoffs' principle [1], an expert in charge of measuring the security level of a technique starts with the assumption that the attacker knows everything except the secret key. It means that the attacker knows how to extract the feature vectors carrying the watermark signal of (6.1). The only thing the attacker knows about the secret key is its sample set (the set of possible key values). We must clarify what the secret is. Almost all techniques create some key-variables (such as the secret direction for spread spectrum, or the dither for quantization index modulation) from a pseudo-random generator seeded with a secret binary word called the "seed." This is a very simple way to restitute these key-variables whenever needed at the embedding and the decoding sides. It is these key-variables which allow anyone to read, write, and erase watermarks. The seed is only an auxiliary variable. The ultimate secret to be disclosed is thus these key-variables.

The first issue of the security assessment is to state whether or not the attacker can disclose some information about this secret key when observing protected contents. This is called "information leakage". The consequence is that the attacker can improve his knowledge about the secret key. At the beginning, he only knows

its sample domain. While carrying on observing protected contents, he can theoretically reduce this set (i.e., refine his estimation of the secret key). The second issue is the speed of this accumulation of knowledge as a function of the number of observations.

This theoretical security assessment results in the number of protected contents that the attacker needs to observe for disclosing the secret key up to a given accuracy. This theoretical study is very hard to conclude, and usually the watermarking scheme and the statistics of the host are simplified to the maximum. Another drawback is that this theoretical study may not give any clue about practical algorithms that take real protected contents as input and outputs the estimated secret key.

For this reason, a third issue is to build such an estimator. On the same over-simplified setup, the estimator performance is lower than foreseen because the theory gives a lower bound of its estimation accuracy. This proves the existence of one estimator doing the job within a limited complexity.

6.1.3.3 Classes of Security Attacks

In order to generalize security assessment to many applications, academics have listed typical scenarios of attacks. These categories are based on the type of observations to which the attacker has access in order to refine his knowledge about the secret key. Here is a nonexhaustive list:

- *Watermarked Only Attack:* the observations are pieces of content watermarked with the same technique and secret key. In most security oriented applications this attack is a threat.
- *Known Original Attack:* the observations are pairs of original host and its watermarked version. For example, a movie trailer might not be watermarked, while a copy taken from a blue-ray may be watermarked.
- *Known Message Attack:* the observations are pairs of watermarked pieces of content and their hidden messages. For example, if the embedded message is the copy "status" of a movie, it is obviously known.
- *Chosen Original Attack:* the observations are pairs of watermarked and original contents chosen by the attacker. This happens when the attacker has access to a watermark embedder as a sealed black box.
- *Multiple Watermarked Attack:* the observations are multiple watermarked versions with different messages of some original contents.

- *Chosen Watermarked Attack:* this is another name for the oracle attack (see Section 6.6).

Not all classes listed above have been analyzed; only a few have been studied (watermarked only attack, known message attack, known original attack) for some watermarking schemes (spread spectrum and quantization index modulation). Section 6.3 presents the three main approaches that have been proposed to measure the theoretical level of security of a watermarking scheme. Section 6.4 lists the main algorithms used to estimate secret keys in practice.

6.1.3.4 The Remaining Uncertainties

An information leakage about the secret key does not mean that the attacker will eventually disclose the secret key. For a multibit watermarking scheme, the watermarked only attack and the known original attack usually estimate the secret up to some uncertainties. In spread spectrum or quantization index modulation, for example, this allows the attacker to read the hidden symbols up to a permutation over their alphabet. In other words, this uncertainty prevents the attacker from embedding and decoding hidden messages. Yet, he can notice when two pieces of content are watermarked with different messages, he can flip some bits (not knowing which of them), and most importantly he can erase the watermark signal without introducing too much distortion.

Cayre and Bas [2] have built categories of watermarking schemes based on their remaining uncertainties under a watermarked only attack. They have also designed two variants of spread spectrum (called natural and circular watermarking) offering much more remaining uncertainties than the original scheme. Therefore, they can be considered more secure [3].

6.2 EXAMPLES OF WORST-CASE ATTACKS

This section details worst-case attacks against the two most well-known watermarking schemes: spread spectrum (see Section **??**) and quantization index modulation (see Section **??**).

6.2.1 Spread Spectrum

References about worst-case attacks against spread spectrum watermarking include the works of Le Guelvouit and Pateux [4], Su and Eggers [5], and Moulin et

al. [6, 7]. Their analyses are similar, but with different flavors, because they assume different host distributions (white or correlated), performance metric P (BER or capacity), and attack distortion metric D_a (MSE or weighted MSE). They typically use the Lagrange multiplier method to minimize P at a given D_a, which ends up with a worst-case attack being a mixture of three strategies as detailed below.

Example: Spread spectrum with Gaussian host samples [4].
Suppose the following model: Host samples are statistically independent and Gaussian distributed with their own variance: $\mathbf{c}_{\mathrm{o}}[j] \sim \mathcal{N}(0, \sigma[j]^2)$. The embedding is as follows

$$\mathbf{c}_{\mathrm{w}}[j] = \mathbf{c}_{\mathrm{o}}[j] + \frac{\mathbf{s}[j]}{\sqrt{N}} \sum_{i=1}^{P} b_i \mathbf{w}_i[j] \tag{6.3}$$

with $b_i \in \{-1, 1\}$ the antipodal modulation of the ith bit to be embedded, $\{\mathbf{w}_1, \ldots, \mathbf{w}_P\}$ the orthonormal secret carriers modulating one bit each (i.e., $\mathbf{w}_k^\top \mathbf{w}_\ell = \delta_{\{k=\ell\}}$), and $\mathbf{s}[j] \geq 0$ a perceptual shaping weight. Then the worst-case attack can be written as $\mathbf{c}_{\mathrm{z}}[j] = \gamma[j]\mathbf{c}_{\mathrm{w}}[j] + \mathbf{n}[j]$, with $\gamma[j] \geq 0$ a scaling factor and $\mathbf{n}[j] \sim \mathcal{N}(0, \rho[j]^2)$. This creates the attack distortion

$$D_a = \mathbb{E}(\|\mathbf{c}_{\mathrm{z}} - \mathbf{c}_{\mathrm{o}}\|^2) = \sum_{j=1}^{N} \sigma[j]^2(1 - \gamma[j])^2 + P\gamma[j]^2\mathbf{s}[j]^2 + \rho[j]^2 \tag{6.4}$$

The optimal expression of the parameters $(\gamma[j], \rho[j]^2)$ can be derived from three strategies:

- *Erasure:* If the jth sample carries a lot of watermark power compared to the power of the host signal, canceling this coefficient lowers the watermark to noise power ratio while creating a small distortion: $(\gamma[j], \rho[j]^2) = (0, 0)$.
- *Wiener filtering:* If the jth sample carries a small watermark power compared to the power of the host signal, estimating the host sample by a Wiener filtering lowers the distortion while not modifying the watermark to noise power ratio: $(\gamma[j], \rho[j]^2) = (\gamma_W[j], 0)$ with

$$\gamma_W[j] = \frac{\sigma[j]^2}{\sigma[j]^2 + P\mathbf{s}[j]^2} \tag{6.5}$$

- *SAWGN (Scale and Add White Gaussian Noise):* In between these two previous cases, the optimal $(\gamma[j], \rho[j]^2)$ are

$$\gamma^\star[j] = \frac{\lambda\sigma[j]^2 - \mathbf{s}[j]}{\lambda \mathbf{s}[j]^2}, \quad \rho^\star[j]^2 = \gamma^\star[j]\left(\gamma_W[j] - \gamma^\star[j]\right)\left(\sigma[j]^2 + P\mathbf{s}[j]^2\right) \tag{6.6}$$

  where λ is a Lagrange multiplier.

The attacker plays the following game: he starts setting λ to a very small value, such that $\gamma^\star[j] < 0$ for $\forall j, 1 \leq j \leq N$. This means that all coefficients should be erased, producing a maximum of total distortion. Then, by slowly increasing λ, it is possible to apply the SAWGN strategy to the indices where both $\gamma^\star[j]$ and $\rho^\star[j]^2$ have positive values, reducing the total distortion $D_a(\lambda)$. By increasing λ again, some indices will see $\rho^\star[j]^2 < 0$ but $\gamma^\star[j] > 0$, so that the best strategy becomes the Wiener filtering. The attacker will stop when $D_a(\lambda)$ is lower or equal to its distortion budget. In the end, depending on this budget and the model's parameters $\{\sigma[j]^2, \mathbf{s}[j]\}$, the worst case attack is a mix of three strategies: erasure, Wiener filtering, and SAWGN.

6.2.2 Quantization Index Modulation (QIM)

A nonexhaustive list of papers on the worst-case attack against QIM with cubic lattices, such as DC-DM (distortion-compensation dither modulation) or scalar Costa scheme (see Section **??**), is [8, 9, 10], and against QIM with arbitrary good lattices [11].

Example: Binary QIM with scalar quantizer and the "3 delta attacks" [8]. The embedding of bit 0 (resp. 1) uses the codebook $\mathcal{C}_0 = \mathbb{Z}\Delta + \boldsymbol{\delta}_K$ (resp. $\mathcal{C}_1 = \mathbb{Z}\Delta + \Delta/2 + \boldsymbol{\delta}_K$), where $\boldsymbol{\delta}_K$ is the secret dither. The distortion compensation is α (see (**??**)) and to gain some robustness we assume $1/2 \leq \alpha \leq 1$. The decoding uses the union of codebook $\mathcal{C}_0 \cup \mathcal{C}_1$. A decoding error happens if $|\mathbf{c}_\mathrm{z}[j] - Q_b(\mathbf{c}_\mathrm{z}[j])| > \Delta/4$ when bit b is embedded in the jth host sample.

If the attacker knows (Δ, α) but not the secret dither $\boldsymbol{\delta}_K$, the worst-case attack restricted on the addition of noise $\mathbf{c}_\mathrm{z}[j] = \mathbf{c}_\mathrm{w}[j] + \mathbf{n}[j]$ consists in drawing independent and identically distributed noise samples according to the probability mass function:

$$\mathbb{P}(\mathbf{n}[j] = -T) = \mathbb{P}(\mathbf{n}[j] = T) = A, \quad \mathbb{P}(\mathbf{n}[j] = 0) = 1 - 2A \tag{6.7}$$

The expected attack distortion per sample is $d_a = D_a/N = 2AT^2$. For a given d_a, the parameters maximizing the BER are

$$T^\star = \begin{cases} \Delta\left(\alpha - \frac{1}{2}\right) & \text{if } \frac{1}{2} \leq \alpha \leq \frac{5}{6} \\ \frac{\Delta}{2}\left(\frac{3}{2} - \alpha\right) & \text{if } \frac{5}{6} \leq \alpha \leq 1 \end{cases} \quad \text{and } A^\star = \frac{d_a}{2T^{\star 2}} \tag{6.8}$$

producing the maximum BER:

$$\text{BER} = \begin{cases} \min\left(\frac{d_a}{\Delta^2}\frac{1}{2(1-\alpha)\left(\alpha-\frac{1}{2}\right)}, 1\right) & \text{if } \frac{1}{2} \leq \alpha \leq \frac{5}{6} \\ \min\left(\frac{d_a}{\Delta^2}\frac{4}{\left(\frac{3}{2}-\alpha\right)^2}, 1\right) & \text{if } \frac{5}{6} \leq \alpha \leq 1 \end{cases} \tag{6.9}$$

The worst-case attack depends on the embedding scheme and its parameters denoted by θ_e. In the previous example, the optimal noise distribution depends on $\theta_e = (\Delta, \alpha)$. Under an attack of parameter θ_a, the performance is given by $P(\theta_e, \theta_a)$. Therefore, a natural question arises: what is the best embedding setup θ_e in the sense that its lowers the impact of the induced worst-case attack? This defines a game between the watermark designer and the attacker where these actors have distortion budgets D_e and D_a, respectively:

$$\max_{\theta_e : D_e(\theta_e) \leq D_e} \min_{\theta_a : D_a(\theta_a) \leq D_a} P(\theta_e, \theta_a) \tag{6.10}$$

The answer for spread spectrum for colored host samples results in the rule of thumb called "power spectrum condition": the power spectrum of the watermark signal should be proportional to the power spectrum of the host [5]. Indeed, the situation is much more involved and cannot be detailed here. The following citations are the most detailed articles on this topic, assuming different host distributions and metrics for performance and distortion [5, 4, 6, 7].

Example: Binary QIM with scalar quantizer and the "3 delta attacks" [8]. Instead of fixing (Δ, α), as in the previous example, we suppose that the watermark fulfills a constraint on the expected embedding distortion per sample: $d_e = \alpha^2\Delta^2/12$. Then the optimal embedding parameters are

$$\alpha^\star = \frac{2}{3} \quad \text{and} \quad \Delta^\star = \sqrt{27 * d_e} \tag{6.11}$$

yielding a maxmin BER of $\min(d_a/3d_e, 1)$. In other words, the "3 delta attack" kills the watermark channel (i.e., BER $= 0.5$) with an attack distortion only $3/2$ times bigger than the embedding distortion.

6.3 HOW TO MEASURE WATERMARKING SECURITY

This section details three different ways to measure the security levels of a watermarking scheme. In this context, security attacks define more the way the attacker steals knowledge about the secret key rather than the exploitation of this stolen knowledge to decode, embed, modify or remove watermarks without authorization.

6.3.1 Fisher Information

This approach considers the secret key as fixed and the observations $o^{N_o} = \{o_1, \ldots, o_{N_o}\}$ as random variables whose distribution is denoted by $f(o; K)$. This makes sense under the watermark only attack; for example, not knowing the host content, the watermarked content appears to be a random variable. The embedding transforms the distribution of the host into the distribution of the watermarked content, which depends on the secret key. The goal of the attacker is to estimate this parameter of the distribution. Hidden messages and perceptual masks might be modeled as nuisance parameters.

The attacker cannot estimate the secret key if the problem is not identifiable: Suppose that $\forall (K', K) \in \mathcal{K}_0 \times \mathcal{K}_0$, $f(o; K') = f(o; K)$. Just by analyzing the observations, the attacker may disclose the set of keys $\mathcal{K}_0$, but this does not grant him the power to uniquely identify which key is the true secret key K inside this subset.

Suppose now that the estimation problem is identifiable. The works [12, 13] use the Cramer-Rao bound to measure watermarking security. Any unbiased estimator $\hat{K}$ of the secret key computed from a sample of N_o independent observations has a covariance matrix (reflecting the estimation noise) $\mathbf{R}_{\hat{K}} \geq N_o^{-1} M_{FI}(K)^{-1}$, in the sense of nonnegative definiteness of the difference matrix. $M_{FI}(K)$ is the Fisher Information Matrix defined as

$$M_{FI}(K) = \mathbb{E}[\boldsymbol{\psi}(K)\boldsymbol{\psi}(K)^{\perp}], \quad \text{with } \boldsymbol{\psi}(K) = \nabla_K f(o; K) \tag{6.12}$$

The mean square error $\mathbb{E}[\|\hat{K} - K\|^2]$ equals the trace of $\mathbf{R}_{\hat{K}}$, its lower bound decreases to zero as N_o^{-1} with a proportional constant $Tr\left(M_{FI}(K)^{-1}\right)$. This last constant is proposed as a measurement of watermarking security in [12]. The more information about the secret key that leaks from the observations, the lower the security level. The main critics are the following:

- Vector $\boldsymbol{\psi}(K)$ has an expression if distribution $f(o; K)$ is derivable with respect to K. This is not possible when K is a discrete variable.

- This measurement of watermarking security fails to capture the impact of remaining uncertainties. Indeed, they often turn the Fisher Information Matrix noninvertible.

6.3.2 Equivocation

A second attempt to define security in watermarking has been a translation of the seminal work proposed by Shannon [14] for cryptography during the 40s to the world of watermarking. The analogy is the following: like a crypto-system producing cipher texts that are functions of the clear texts and the secret key, a watermarking embedder produces watermarked contents from messages to be hidden, original contents, and the secret key.

The attacker knows nothing about the secret key K except its sample domain $\mathcal{K}$ and the way it has been generated, which amounts to its distribution p_K (a probability mass function if K is discrete, or a probability density function if it is continuous). This motivates the assumption that the secret key is a random variable K for the attacker. The entropy of this variable measures the amount of the attacker's uncertainty[1]: $H(K) = -\int\!\!\!\!\!\!\sum_{\mathcal{K}} p_K(k) \log p_K(k)$, expressed in bits if the logarithm is to the base of 2.

Depending on the class of the attack (see Section 6.1.3.3), the attacker has N_o observations of a given nature, denoted as $o^{N_o} = \{o_1, \ldots, o_{N_o}\}$. This transforms the a priori distribution $p_K(k)$ into a posteriori distribution $p_K(k|o^{N_o})$. The equivocation, $H(K|O^{N_o}) = \mathbb{E}_{O^{N_o}}[H(K|o^{N_o})]$ measures the amount of the remaining uncertainty from the attacker point of view. We are interested in how it decreases with the number of observations and we define $h(N_o) = H(K|O^{N_o})$ for $N_o \geq 1$, and $h(0) = H(K)$.

6.3.2.1 Discrete Random Variable

The secret key is a discrete random variable. Then, equivocation $h(N_o)$ is a nonnegative and nonincreasing function. It converges to a minimum value h_∞ as $N_o \to \infty$. If this minimum is $h_\infty = 0$, it means that by carrying on observing data, the attacker will gather enough information to uniquely determine the secret key K. If not, the attacker will reduce the key space $\mathcal{K}$ to a subset of size at least 2^{h_∞} (when the equivocation is expressed in bits). If the equivocation is a constant function (i.e.,

1 The expression $\int\!\!\!\!\!\!\sum_{\mathcal{X}} g(x)$ means $\sum_{x \in \mathcal{X}} g(x)$ if X is a discrete variable ($\mathcal{X}$ is finite or countable) and $\int_{\mathcal{X}} g(x) \partial x$ if X is continuous.

$h(N_o) = H(K)$), it proves that observations bring no information about the secret key. The watermarking scheme is then perfectly secure.

Example: Substitution scheme with binary data [12].
Vector $\mathbf{c}_\mathrm{o}$ is a binary word of length N. The secret key is a P-uple containing distinct indices in $\{1, \cdots, N\}$: $K = (k[1], \cdots, k[P])$. The embedding substitutes some bits in the host cover, whose indices are given by the secret key, by message bits:

$$\mathbf{c}_\mathrm{w}[k[i]] = b[i] \quad \forall 1 \leq i \leq P, \text{ otherwise } \mathbf{c}_\mathrm{w}[j] = \mathbf{c}_\mathrm{o}[j] \tag{6.13}$$

We denote by $\mathbf{c}_\mathrm{w}^{(K)}$ the restriction of $\mathbf{c}_\mathrm{w}$ over the set of indices given by K s.t. we can rewrite the embedding as $\mathbf{c}_\mathrm{w}^{(K)} = \mathbf{b}$. Let us compute $H(K)$ first. The secret key, as a discrete random variable, is uniformly distributed. There are $|\mathcal{K}| = N!/(N-P)!$ possible values. Therefore, $p_K(k) = |\mathcal{K}|^{-1}$, and

$$H(K) = \sum_{k \in \mathcal{K}} -p_K(k) \log p_K(k) = \log |\mathcal{K}| \tag{6.14}$$

Watermarked Only Attack. The observations denoted by o^{N_o} are N_o random watermarked content. The message to be hidden $\mathbf{b}$ is uniformly distributed with probability $1/2^P$. We have:

$$p(\mathbf{c}_\mathrm{w}|K) = \sum_{\mathbf{b}} p(\mathbf{c}_\mathrm{w}|\mathbf{b}, K) p(\mathbf{b}|K) = \sum_{\mathbf{b}} p(\mathbf{c}_\mathrm{w}|\mathbf{b}, K) p(\mathbf{b}) \tag{6.15}$$

$$= \sum_{\mathbf{b}} \frac{\delta_{[\mathbf{c}_\mathrm{w}^{(K)} = \mathbf{b}]}}{2^{(N-P)}} \frac{1}{2^P} = \frac{1}{2^N} \tag{6.16}$$

and also

$$p(\mathbf{c}_\mathrm{w}) = \sum_K p(\mathbf{c}_\mathrm{w}|K) p_K(K) = \frac{1}{2^N} \sum_K p_K(k) = \frac{1}{2^N}. \tag{6.17}$$

Therefore, by Bayes rule, $p_K(k|\mathbf{c}_\mathrm{w}) = p_K(k)$, which implies that $I(K; o^{N_o}) = 0$ and $h(N_o) = H(K)$ for all $N_o > 0$. This shows that no information about the secret key leaks from watermarked contents.

Known Original Attack. The situation is less secure as $h(N_o)$ decreases and converges to $\log_2(P!)$ (see [12]). This amounts for the remaining uncertainty: the attacker eventually discloses the indices of K, but not their ordering. He cannot read

or write messages, yet, he can observe whether two contents share the same hidden message, or he can modify the hidden message (not knowing what he is writing).

Known Message Attack. The situation is even less secure as $h(N_o)$ decreases and converges to zero asymptotically (see [12]). Within $N_o \approx \log_2(PN)$ observations he has theoretically enough information for uniquely identifying the secret key.

6.3.2.2 Continuous Random Variable

The interpretation of the equivocation is less straightforward in this case. Equivocation $h(N_o)$ is a nonincreasing function, but it can be negative. Pérez-Freire and Pérez-González [15] give a complete analysis under this framework for spread spectrum like schemes with and without side informed embedding, whereas Pérez-Freire et al. [16, 17] cover quantization index modulation.

Example: Additive spread spectrum without side information [15].
The embedding is described in (6.3), where $P = 1$. The secret K generates a secret carrier $\mathbf{w}$ modeled as a white Gaussian vector of variance $\sigma_w^2 = 1/N$. The host cover follows the same distribution with variance σ_o^2, while the perceptual shaping weight is assumed to be constant: $\mathbf{s}[j] = s$. Observing independent samples $\{\mathbf{c}_{\mathrm{w_i}}, b_i\}_{i=1}^{N_o}$, the best estimator is:

$$\hat{\mathbf{w}} = \frac{\sigma_w^2}{\sigma_o^2 + N_o\sigma_w^2} \sum_{i=1}^{N_o} b_i \mathbf{c}_{\mathrm{w_i}} \tag{6.18}$$

in the sense that $\hat{\mathbf{w}}$ follows the same distribution as $\mathbf{w}$. This produces the following equivocation:

$$h(N_o) = \frac{N}{2} \log\left(2\pi e \frac{\sigma_w^2\sigma_o^2}{\sigma_o^2 + N_o\sigma_w^2}\right) \tag{6.19}$$

A interesting interpretation of the equivocation for continuous secret signal makes the connection with estimation theory [15]. Denote by σ_e^2 the variance per dimension of the attacker's estimation of the secret. The equivocation gives a lower bounds of this variance:

$$\sigma_e^2 \geq \frac{1}{2\pi e} \exp\left(\frac{2}{N} h(N_o)\right) \tag{6.20}$$

By using (6.19) and for large N_o, we find back the Cramer-Rao bound mentioned in Section 6.3.1 for additive spread spectrum. This lower bound holds whatever the estimator used by the attacker. However, due to the remaining uncertainties (especially under WOA), the accuracy of the estimator can be much bigger.

6.3.3 Effective Key Length

Watermarking and cryptography are two security primitives. It is not surprising that the previous security analysis framework is deeply inspired by the similar work Shannon did in cryptanalysis. However, there is a big difference.

The notion of *estimated secret key* does not make sense in cryptography. The attacker must find the unique key that decodes cipher texts. It is an "all or nothing" game. In watermarking, one might be able to read, write, or erase watermarks with an estimated key, which is not exactly the secret key. Of course, the more accurate the estimation, the more reliable the access to the watermarking channel.

Example: Spread spectrum scheme with binary watermark signal [18].
Suppose that the secret direction $\mathbf{w}$ is a vector of N components taking ± 1 values. In other words, $\mathcal{K} = \{-1, +1\}^N$. There are $|\mathcal{K}| = 2^N$ possible secret directions. Finding this secret key over of such a large set has a complexity equaling N in logarithmic scale. Suppose now that an attack is successful, provided that the attacker finds a key close to the secret direction; their correlation with the true secret direction equals ρN, with $-1 \leq \rho \leq 1$. This means that the estimated and secret keys agree on $N_+ = N(1+\rho)/2$ components. There are indeed $\binom{N}{N_+}$ keys in $\mathcal{K}$ meeting this constraint. Finding one of these in $\mathcal{K}$ has a logarithmic complexity $\approx N(1 - h_2((1+\rho)/2))$ in logarithmic scale (asymptotically as $N \to \infty$, thanks to the Stirling formula), where $h_2(p)$ is the entropy in bits of a random Bernoulli binary random variable X s.t. $\mathbb{P}(X = 1) = p$. If $\rho = 0.4$, the logarithmic complexity is $0.12N$ bits, which is a much smaller security level than N bits.

The framework presented in Section 6.3.2 assumes that the aim of the attacker is to disclose the key used at the embedding side thanks to the observations. Yet, the last section shows a pitfall in this approach: the decoding key can be different from the embedding key. How accurate should the estimated key be to successfully hack the watermarking scheme? This section details a new framework that was proposed recently. It aims to jointly consider the estimation of the secret key with its use in a worst case attack.

First, it starts by defining what a successful attack is. For example, the attacker is willing to decode hidden messages with a given probability of success

$1-\eta$. Second, it turns this requirement into a set of keys, called the "equivalent keys" which achieve this goal. This set depends on the true secret key K and the requirement η, and is denoted $\mathcal{K}_{eq}(K,\eta)$. A successful attack is equivalent to the disclosure of one of these equivalent keys.

We now give the attacker a random sample of N_o observations, from which he derives an estimation $\hat{K}$ of the secret key. This is a random variable because the sample is random. We now compute the probability that this estimation is one of the equivalent keys: $\mathbb{P}(\hat{K} \in \mathcal{K}_{eq}(K,\eta))$. The *effective key length* measured in bits is defined by:

$$L \triangleq -\log_2 \mathbb{P}(\hat{K} \in \mathcal{K}_{eq}(K,\eta)) \tag{6.21}$$

This quantity measures '*the inability by an unauthorized users to have access to the raw watermarking channel,*' which is the way Kalker defined watermarking security in [19]. This also strengthens the analogy with cryptography: $1/2^L$ is the probability that a key randomly picked by a brute force attack is indeed the unique secret key, when keys are sequences of L bits.

Example: Decoding of spread spectrum with binary watermark signal.
We consider the same setup as in the previous example. We assume that the host samples are i.i.d. as $\mathcal{N}(0,\sigma^2)$ and the embedding of one bit is done as follows: $\mathbf{c}_{\mathrm{w}} = \mathbf{c}_{\mathrm{o}} + \frac{s}{\sqrt{N}} b\mathbf{w}$. The bit error rate for the legitimate decoder, which uses the secret direction $\mathbf{w}$, is given by:

$$\mathrm{BER}_D = \Phi\left(-\frac{s}{\sigma}\right) \tag{6.22}$$

Suppose that the attacker is now willing to decode the hidden bit using an estimated key $\mathbf{w}' \in \mathcal{K}$ s.t. $\mathbf{w}^\top \mathbf{w}' = N\rho$. This leads to the following $\mathrm{BER}_A = \Phi(-\rho\frac{s}{\sigma})$. The attack is deemed successful if $\mathrm{BER}_A \leq \eta$, which implies that $\mathbf{w}'$ is an equivalent key if $\rho \geq \rho_{\min} = \Phi^{-1}(\eta)/\Phi^{-1}(\mathrm{BER}_D)$. Contrary to the previous example, an equivalent key has a normalized correlation that is greater than or equal to a lower bound $\rho_{\min}$. If the attacker is randomly sampling $\mathcal{K}$, the probability of picking an equivalent key is the probability that a random variable distributed as a Binomial distribution $\mathcal{B}(N,1/2)$ is bigger than $N(1+\rho_{\min})/2$, which is approximately, for large N, $\approx \Phi(-\rho_{\min}\sqrt{N})$. This gives an effective key length in the order of

$$L \approx \frac{\rho_{\min}^2}{2\log(2)} N \tag{6.23}$$

As a consequence, for $\text{BER}_D = 10^{-3}$ and $\eta = 10^{-1}$, we have $\rho_{\min} \approx 0.4$, so that $L \approx 0.115 * N$ bits. Again, we find back that the security level is proportional to N but with a small proportional constant.

Example: Detection of spread spectrum with binary watermark signal.
We consider the same setup as in the previous example but under a detection framework. The embedding is simply: $\mathbf{c}_{\text{w}} = \mathbf{c}_{\text{o}} + s_e \mathbf{w}\sqrt{N}$. At the detection side, the threshold τ is set to meet a requirement on the probability of false alarm: $\tau = \sqrt{N}/\sigma\Phi^{-1}(1 - P_{\text{fa}})$. This fixes the probability of a miss detection: $P_{\text{miss}} = \Phi(\Phi^{-1}(1 - P_{\text{fa}}) - s_e/\sqrt{N}\sigma)$.

The goal of the attacker is to remove the watermark signal: $\mathbf{c}_{\text{z}} = \mathbf{c}_{\text{w}} - s_a \mathbf{w}'/\sqrt{N}$, with an attack distortion which is ν times more than the embedding distortion (i.e., $s_a = \sqrt{\nu} s_e$) and a probability of success of at least $1 - \eta$. This gives a constraint on the equivalent key: its normalized correlation with the true secret key $\mathbf{w}$ must be s.t.

$$\rho \geq \rho_{\min} = \frac{s_e}{s_a} \cdot \frac{\Phi^{-1}(1-\eta) - \Phi^{-1}(P_{\text{miss}})}{\Phi^{-1}(1 - P_{\text{fa}}) - \Phi^{-1}(P_{\text{miss}})} \tag{6.24}$$

Consequently, for $P_{\text{fa}} = 10^{-6}$, $P_{\text{miss}} = 10^{-1}$, $\eta = 10^{-1}$ and $\nu = 2$, using approximation (6.23), we have $L \approx 0.06 * N$.

Both examples above assume a basic and simple model, especially because the attacker picks an estimated key in $\mathcal{K}$, whereas the general framework grants him N_o observations to increase the probability of picking an equivalent key. The analysis is then more cumbersome. We refer the reader to the following papers dealing with spread spectrum like schemes [20, 21] or QIM schemes [22].

To conclude this section, Fisher information, equivocation, and effective key length are not the only ways to gauge watermarking security. The other approaches, like [23], have been somehow less investigated.

6.4 ALGORITHMS AND TOOLS

The previous section surveyed approaches for quantifying watermarking security. For simple models, theoretical developments give close form expressions or bounds of the above-mentioned quantities. This section briefly reviews signal processing or machine learning tools that have been used to disclose the secret key in practice.

6.4.1 Spread Spectrum Like Schemes

The main principle of spread spectrum is to focus the watermark power in a subspace of small dimension, spanned by the secret carriers. The attacker leverages this principle by identifying the subspace of higher energy thanks to a principal component analysis (PCA) [24] or an independent component analysis (ICA) [12]. This works well when the watermark signal is independent from the host, but also against side informed embedding to some extend [25].

This kind of scheme usually produces watermarked signals deeply located inside decoding regions. Therefore, these signals are all concentrated along several directions of the space. The attacker leverages this pitfall by using clustering algorithms like k-means [25, 26, 27]. This idea has been pushed further with a total variational approach, minimizing a cost function modeling this concentration phenomenon by a conjugate gradient distance in the case of the ISS watermarking scheme [15].

6.4.2 Quantization Index Modulation

QIM schemes are also attacked with the principle that watermarked signals are packed around lattice codewords. Thanks to the periodicity of the lattice codewords, a lattice modulo operation folds the space such that the secret dither lies in a finite region around each watermarked signal (under the KMA or CMA scenarios). Set membership algorithms can then compute an approximate intersection of these feasible regions, which shrinks around the secret dither as the attacker analyzes more observations [16]. The attack is more involved under the watermarked-only attack (WOA) scenario [17].

6.5 WHAT WE KNOW SO FAR ABOUT WATERMARKING SECURITY

The following sections sum up the results known so far.

6.5.1 Differences Between WOA and KMA

The WOA is more difficult because the attacker has access to less data. WOA has two limitations:

- The estimation of the secret is harder. However, the difference when compared to KMA (known message attack) vanishes as the hidden message length is small.
- There are remaining uncertainties under WOA. Even after disclosing all parts of the secret, the attacker can't read or write a message because he does not know which part of the secret is coding which bit. He can, however, see if two hidden messages are the same; he can flip a bit (whose position in the hidden message is unknown) or remove the watermark signal.

6.5.2 Trade-Off Between Robustness and Security

Usually, for a given embedding scheme, there is a trade-off between robustness and security. There are some exceptions: under some specific setups, improved sprectrum spectrum (ISS) and correlation aware spread spectrum (CASS [28]) may witness a decrease of the security level as well as a decrease in robustness when fine-tuning their parameters for a given embedding distortion; see [15] or [20].

This comment holds for a given watermarking scheme. Yet, as far as we know, there is no theoretical analysis of a would-be optimum trade-off between security and robustness.

6.5.3 Orders of Magnitude

For spread spectrum like watermarking, the attacker needs some hundreds of watermarked contents when $N \approx 100$, and some thousands of them when $N \approx 100,000$ (for a given embedding distortion budget), to disclose the secret directions when a dozens of bits are hidden.

For quantization index modulation schemes, the security level may vary a lot; it can be very high if robustness is not an issue, but very low as soon as a good robustness is achieved. In this latter case, the security level is much lower than spread spectrum techniques, and ≈ 10 observations might be enough to disclose the secret dither.

These orders of magnitude are provided under the assumption that the attacker has access to the cover samples carrying the watermark signal.

6.6 ORACLE ATTACKS

In an oracle attack, the attacker has an unlimited access to a watermark detector or decoder enclosed in a sealed black box. The attacker has one of the three following goals:

1. To remove the watermark of a protected content.
2. To hide a message or to overwrite the message already hidden in a content.
3. To disclose a part of the secret key.

The first two goals are conceptually identical as soon as we imagine the set of all possible contents. This huge ensemble can be partitionned into regions of contents producing the same decoding / detection output. The attacker has a content not belonging to the desired decoding region. The aim is to shift it into another region. This is the region of unwatermarked content in goal (1), or the region of content hiding the desired message in goal (2). The attacker would like to find the closest content on the other side of the frontier enclosing the targeted region. We called this attack a *closest point attack* and it belongs to the category of worst-case attack of Section 6.1.2.

The third goal is totally different. We assume that any protected content is watermarked with the same secret key K. This secret parameter is also embedded in the sealed black box. The attacker feeds the decoder with contents and saves their outputs. These observations, pairs of content / output, may leak information about K [29]. We call this process the *chosen watermarked attack*, which pertains to the category of security attacks of Section 6.1.3.

6.6.1 Sensitivity Attack

Although chasing different goals, closest point attacks and chosen watermarked attacks resort to the same core process: the sensitivity attack, whose roots date back to [30]. The attacker "works" with extracted features whose space $\mathcal{F}$ is of dimension n_f, in the sense that he is able to modify a content such that its extracted features equal a given target. These features may not be the same as the ones used by the embedder for carrying the watermark signal in (6.1).

A naive oracle attack submits content whose feature vector $\mathbf{f}$ samples the feature space $\mathcal{F}$. This can be done by probing over a regular grid. The attacker creates a map of $\mathcal{F}$ with white points when the decision of the decoder is the desired one and black points otherwise. Having disclosed this map, he knows

how to get outside the black region in order to reach the white region with the shortest path (i.e., with the minimum distortion). If the grid has a step Δ and $\overline{F}$ denotes the typical amplitude of features, then there are $(\overline{F}/\Delta)^{n_f}$ points to be tested. This is exponential with the number of features, and it does not lead to a tractable oracle attack in practice. The detector output discloses, on average, very little information. During the naive oracle attack, the attacker would see some long series of constant outputs. The useful information is indeed located when sampling near the frontier where the detector outputs change. Disclosing regions or their frontiers are equivalent problems. This is the goal of the sensitivity attack.

The first step in this attack is to find a point on the frontier. This is called a "sensitive content" because a small perturbation flips the detection output with a good probability. To find a sensitive vector, the attacker needs two pieces of content whose feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ are not located in the same region. For goal (1), the attacker strongly distorts a protected content until it is deemed "nonwatermarked." For goal (2), he needs at least one piece of content watermarked with the desired hidden message. In the feature space $\mathcal{F}$, the line going from $\mathbf{f}_1$ to $\mathbf{f}_2$ intersects the frontier. A dichotomy line search repeatedly submits to the sealed black box content whose extracted feature is along this line, and according to the output, it will iterate and converge to the sensitive content.

Once a sensitive vector is found, the second step adds a small random perturbation to its features and submits the modified vector to the detector. By repeating this process, the attacker is then able to have a local approximation of the frontier in $\mathcal{F}$, only valid in the neighborhood of the sensitive point. This approximation is up to the first order: the frontier is approximated by a tangent hyperplane. Denote by $F_{\mathcal{F}}(.)$ the function describing the decoding region as $\{\mathbf{f} \in \mathcal{F} | F_{\mathcal{F}}(\mathbf{f}) > 0\}$. The normal vector of the hyperplane is the gradient of this function. This requires $O(n_f)$ iterations. Up to the second order, the attacker approximates the frontier by a quadric surface defined by the gradient and the Hessian of $F_{\mathcal{F}}(.)$. This requires $O(n_f^2)$ iterations.

6.6.2 Closest Point Attack

A first use of the sensitivity attack is the closest point attack (CPA): $\mathbf{f}_1$ is the feature vector of the content to be hacked, $\mathbf{f}_2$ is lying on the targeted region, but this content is too perceptually different. The sensitive content between $\mathbf{f}_1$ and $\mathbf{f}_2$ has a better quality. Once the frontier is locally estimated, the attacker knows in which direction he should push the extracted vector in order to get closer to $\mathbf{f}_1$ while staying close to the frontier. From this new point $\mathbf{f}_3$, the attacker again finds a sensitive vector,

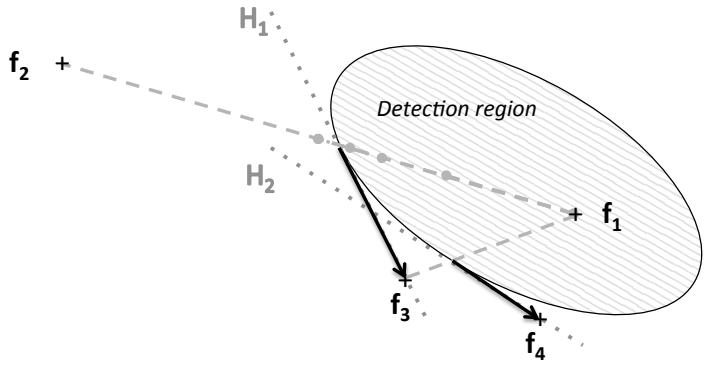

Figure 6.1 The closest point attack with $n_f = 2$. Tangent planes are sketched with dotted lines. The arrow shows the direction of the next move in order to get closer to $\mathbf{f}_1$.

which should be nearer, and approximates again the frontier. This process is iterated until the improvement in quality of the sensitive content is no longer meaningful. Figure 6.1 shows the first iterations of the CPA in $n_f = 2$ dimensions.

This CPA is called BNSA (Blind Newton Sensitivity Attack) by its inventors [31, 32]. Its main advantage is that no assumption at all is needed with respect to the shape of the decoding region. Experimental simulations show that the algorithm quickly converges with the gradient option; around $M = 10$ iterations are needed. This makes the Hessian estimation not worth it at all. The final sensitive content is of very good quality, although some differences exist depending on the watermarking scheme. Some techniques are more robust than others against the BNSA, in the sense that the final attacked content is more degraded. The researchers suspect that some watermarking schemes have bigger detection areas (for a given probability of false alarm) or more efficient embedders so that the watermarked feature vector $\mathbf{f}_1$ is more deeply located inside of the detection area. Another explanation is that the BNSA converges to the global minimum distortion if the decoding region is convex, and to a local minimizer otherwise.

The number of detector trials is $O(Mn_f)$ because, at each iteration, the sensitivity attack estimates the tangent hyperplane. Note that this estimation step can be done in parallel if the attacker has several decoders in hand. This attack works in theory, even if $\mathcal{F}$ is not the embedding domain. Yet, when performed on the spatial domain of real images, it needs millions of detection trials. The attacker should

work with features of low dimension. Knowing that the watermarking technique does not modify some high frequencies' coefficients, for instance, is a crucial piece of information to make the attack work in practice. When not knowing the exact features used by the watermarking scheme, the attacker can play with his own features. If he gathers more features than needed, his attack lasts longer, but features carrying no watermark signal will almost not be distorted by the attack. If he works with too few features, the watermark signal will not be completely erased, and the quality of the final content is poorer.

This is the reason why Earl [33] introduced a notion of perceptual importance of the feature. The attacker performs his attack only with the most perceptually important features that are supposed to carry the biggest part of the watermark energy. This helps strike a better trade-off between the quality of the attacked content and the complexity of the attack. When tested on real images, this CPA needs some thousands of trials. Moreover, Earl proposed a method where the attacker does not have to wait for $O(n_f)$ detection trials (i.e., the end of a sensitivity attack estimating the tangent hyperplane) to submit an intermediate attacked content. However, this CPA cannot be run in parallel with several decoders.

An important feature of the CPA is that the quality of the attacked content keeps improving as the number of detection trials increases, but with an uneven speed. The quality improvement is huge for the first iterations, but then it stalls so that it requires a huge amount of trials to forge a copy with pristine quality.

We remind the reader that the goal of this attack is to remove (or modify) the watermark of a particular piece of content. The attack starts from scratch if another piece of watermarked content is to be attacked.

6.6.3 Chosen Watermarked Attack

The chosen watermarked attack (CWA) is in essence very different from the CPA, as it discloses the secret key, hence it is run only once. Moreover, this security attack needs knowledge of the algorithm, especially the embedding domain and the nature of the decoding regions (quadrants, cones, quadrics...). Formally, suppose that a content belongs to the decoding region if $F(\mathbf{c}, K) > \tau$. The attacker knows the generic function $F(.,.)$, but not the secret key K. With a sensitivity attack, he approximates the frontier around the sensitive vector $\mathbf{c}_\mathrm{s}$ with a tangent plane whose normal direction is the gradient of the function:

$$\mathbf{n}_{\mathbf{c}_\mathrm{s}} = \nabla_{\mathbf{c}_\mathrm{s}} F(\mathbf{c}_\mathrm{s}, k) \tag{6.25}$$

This equation is vectorial, hence, it indeed gathers N scalar equations, and all variables are known except for the secret key. If the secret key is a vector of same length, then the attacker has theoretically enough information to disclose the secret key provided this equation has a unique solution. For instance, for the spread spectrum scheme, $F(\mathbf{c}, k) = \mathbf{c}^\top \mathbf{w}$, where $\mathbf{w}$ is the secret direction (or carrier). It immediately follows that $\mathbf{n}_{\mathbf{c}_\mathrm{s}} = \mathbf{w}$. Estimating the hyperplane gives the normal direction $\mathbf{n}_{\mathbf{c}_\mathrm{s}}$, which, in turn, reveals the secret parameter of the scheme. If the secret key is a $N \times N$ matrix (for instance, JANIS order 2 [34]), the attacker needs N sensitivity attacks yielding each N independent equations, for a total of $O(N^2)$ detector trials.

The CWA has only been studied with zero-bit watermarking schemes. Extension to multibit technique should be straightforward. The seminal paper is due to Kalker, Linnartz and van Dijk [35]. A more elaborated study is [36].

6.6.4 Countermeasures

The countermeasures proposed so far mainly focus on the core process of the oracle attacks: the sensitivity attack.

A first idea is to slow down the watermark detection. For example, Blu-ray disc players wait for 20 minutes before shutting down the playback of a copy deemed as pirated due to the presence of a watermark. It lets malicious users enjoy 20 minutes of content, but on the other hand, an oracle attack becomes impractical.

A second idea is to randomize the detection output in order to spoil the estimation of the tangent hyper-plane. Its success is mitigated; the sensitivity attack still works, but it needs more detection trials. Geometrically, a gray area is inserted along the frontier as sketched in Figure 6.2. In this gray area, the detection output is random: the detector flips a coin to take a decision. However, the frontier of the detection region does not change. The attacker needs to know on which side of the frontier a feature $\mathbf{c}_\mathrm{s}$ lies. Hence, this feature vector is tested a few times to disclose whether the output is deterministic (always positive in detection region), or random (in the gray area). The number of detection trials is multiplied by a small constant, and it is still linear in N [37].

A third idea is to render the frontier more chaotic or less soft, so that the tangent plane is greatly changing or not even mathematically defined in the neighborhood of the sensitive content, see Figure 6.2. However, the decoding region cannot be fully chaotic because we need to accurately estimate the probability of a false alarm. Therefore, some researchers propose to start with a well-studied shape

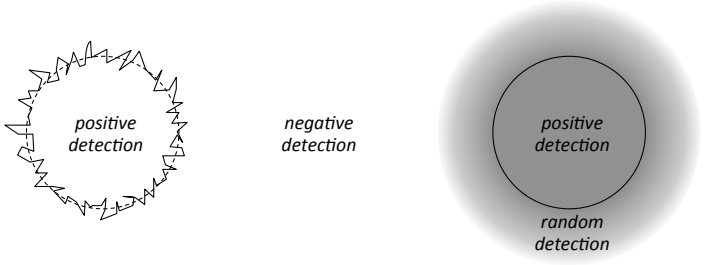

Figure 6.2 Left: The frontier of the detection region has been "fractalized." Right: The detection output is random when content is located in between positive and negative detection regions.

(i.e., quadrant or cone) whose impact on the probability of false alarm is well-understood, and to locally "fractalize" its frontier [38]. This counter-measure works poorly, as everything is a matter of scale. If the frontier looks chaotic when observed in great detail, it is still a soft surface when zooming out. The sensitivity attack still works, but with a bigger step (sensitive vectors are more distorted). The accuracy of the estimation of the tangent plane decreases, but not drastically.

The same idea can prevent a CPA, not on the estimation of the tangent plane, but on its convergence. If the decoding region is not convex, it converges to a local minimum that may not be the closest point. In its original design, the technique *broken arrow* uses some traps to stop oracle attacks [39]. Yet, it is not clear whether this design was proven useful during the BOWS2 competition [40]. However, this more complex decoding region raises the issue of the probability of a false alarm. Other researchers have reported a loss of robustness [41].

6.6.5 Some Comments on the Oracle Attack

The oracle attacks (CPA and CWA) are working in practice; they have been successful during both editions of the BOWS contest [40, 42]. However, it is a difficult attack in a practical setup. In a copy protection application, the watermark detector is inside a compliant device, like a DVD player. Therefore, in order to make a trial, the attacker has to create content, burn it on a blank disk, insert the disc in the device, and wait for the detection output. This certainly takes too long. The real issue is whether or not the attacker will be able to circumvent these constraints in order to speed up the oracle.

The most efficient countermeasures play with the practical setup. For example, most papers have implicitly considered that the detector is memoryless. A smart

detector with some memory could refuse to give a decision when noticing that an oracle attack is going on [43].

6.7 PROTOCOL ATTACKS

This chapter has mostly assumed that the goal of the attacker is to remove a watermark. Disclosing the secret key also enables the attacker to decode hidden messages and to embed or modify hidden messages, such that an authorized decoder can retrieve them.

There are some other flaws, so-called *protocol attacks*, often stemming from misuses of watermarking technology. There is no general framework encompassing all protocol attacks. We present some of them through three case studies.

Example: The watermark copy attack.
This is the most well-known protocol attack [44]. It simply assumes a very robust watermarking scheme with no side informed embedding (e.g., additive spread spectrum). The watermark signal depends on the original cover only through the perceptual slack. A denoising algorithm is able to strip out a part of watermarking energy from the watermarked content. This part is not enough to remove the presence of the watermark thanks to the high robustness of the technique. Yet, this is not the goal of the attacker. The difference between the watermarked content and its denoised version is a rough estimation of the watermark signal. Inserting it into another content may result in a new watermarked content due to the high robustness of the technique. In other words, the attacker succeeds in copying and pasting the watermark.

Example: Proof of owernship.
Suppose a person claims ownership of the image $\mathbf{i}$ by exhibiting a watermark detector $d(\cdot,\cdot)$, a secret key K, and a threshold τ such that $d(\mathbf{i}, K) > \tau$. In other words, this image triggers the detection of a watermark.

First, we could be convinced that this proves this person has watermarked that image some time ago. It does not imply that this person is the true author or a legitimate copyright holder of this work. Anybody can watermark images with their own technique and secret key.

Second, we must be careful about the threshold τ. It is easy to first compute the score $d(\mathbf{i}, K)$ and then to pick a smaller threshold τ. We must verify the soundness of the value of τ by deriving the probability of false positive: $P_{\text{fa}} =$

$\mathbb{P}(d(\mathbf{i}_o, K) > \tau)$, where $\mathbf{i}_o$ is an unwatermarked image. This probability should be small, but, for the sake of robustness, it cannot be zero. Usually, τ is set such that P_{fa} is in the order of 10^{-6}.

This means that, for a fixed K, over $\lceil 1/P_{\text{fa}} \rceil$ random images, one expects one false positive detection, and this person can claim ownership of that particular image. However, it is unlikely that this random image has some value. On the other hand, it also means that, for a fixed image $\mathbf{i}$, by testing N_K secret keys, the probability of finding at least one key producing a positive detection is $1 - (1 - P_{\text{fa}})^{N_K} \approx P_{\text{fa}} N_K$. One sees that if N_K is in the order of $1/P_{\text{fa}}$, this person likely finds at least one secret key that triggers the detection.

This shows that the watermark detection brings a proof of low value in this context. To strengthen this proof, this person should show that the secret key was randomly drawn before the watermarking, independent of the image to be watermarked. For example, this key is also used to embed watermark in other photos which are known to be his previous works of the artist. As far as copyright protection is concerned, from a legal point of view, an author should belong to a society of authors where he registers his works. The deposit of the secret key to this society is a commitment that will bring trust in the watermark detection later on.

Example: Copy protection.
This example is about the playback of pirated content by the first version of some Blu-ray disc players. The detection of a watermarking in the audio streams of a nonciphered movie warns the Blu-ray player that the movie is pirated (camcorded in a theater, for example). The watermark detection only runs on audio streams encoded with the standard audio format. The format is labelled in the header of the audio stream. By just modifying this label, pirates succeed bypassing the watermark detector. The pirates then ask the player to output the audio stream to an external renderer, such as a PC, which decodes the audio stream without taking care of the corrupted format label in the header. This protocol attack benefits from an implementation flaw which has nothing to do with robustness, worst-case attack, or watermarking security.

6.8 CONCLUSION

Worst-case attacks, security attacks, and oracle attacks are three different concepts that are now well-understood in the watermarking community. However, research

articles proposing new watermarking schemes almost never encompass their analysis. This restricts their use to applications where security is not a requirement.

Most robust watermarking schemes have a weak security level. Designing schemes where security is the top requirement is still in its infancy [3]. The research community is still missing the theoretical optimum trade-off between robustness and security.

References

[1] Kerckhoffs, A., "La cryptographie militaire," *Journal des Sciences Militaires*, Vol. 9, janvier 1883, pp. 5–38.

[2] Cayre, F., and P. Bas, "Kerckhoffs-based embedding security classes for WOA data-hiding," *IEEE Transactions on Information Forensics and Security*, Institute of Electrical and Electronics Engineers (IEEE), Vol. 3, No. 1, Mar. 2008, pp. 1–15.

[3] Mathon, B., et al., "Comparison of secure spread-spectrum modulations applied to still image watermarking," *Annales des Télécommunications*, Springer Verlag (Germany), Vol. 64, No. 11-12, Oct. 2009, pp. 801–813.

[4] Pateux, S., and G. Le Guelvouit, "Practical watermarking scheme based on wide spread spectrum and game theory," *Signal Processing: Image Communication*, Vol. 18, April 2003, pp. 283–296.

[5] Su, J., J. Eggers, and B. Girod, "Analysis of digital watermarks subjected to optimum linear filtering and additive noise," *Signal processing*, Elsevier, Vol. 81, 2001, pp. 1141–1175.

[6] Moulin, P., and A. Ivanovic, "The zero-rate spread-spectrum watermarking game," *Signal Processing, IEEE Transactions on*, Vol. 51, No. 4, Apr 2003, pp. 1098–1117.

[7] Moulin, P., and J. O'Sullivan, "Information-theoretic analysis of information hiding," *Information Theory, IEEE Transactions on*, Vol. 49, No. 3, Mar 2003, pp. 563–593.

[8] Vila-Forcén, J., et al., "Worst case additive attack against quantization-based data-hiding methods," in *Security, Steganography, and Watermarking of Multimedia Contents VII*, Vol. 5681 of *Proceedings of SPIE-IS&T Electronic Imaging*, SPIE, San Jose, CA, USA, 2005, pp. 136–146.

[9] Pérez-González, F., "The Importance of Aliasing in Structured Quantization Index Modulation Data Hiding," in *Digital Watermarking*, Vol. 2939 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 1–17, 2004.

[10] Tzschoppe, R., et al., "Additive non-Gaussian noise attacks on the scalar Costa scheme (SCS)," in *Security, Steganography, and Watermarking of Multimedia Contents VII*, Vol. 5681 of *Proceedings of SPIE-IS&T Electronic Imaging*, SPIE, 2005, pp. 114–123.

[11] Moulin, P., and A. Goteti, "Block QIM watermarking games," *Information Forensics and Security, IEEE Transactions on*, Vol. 1, No. 3, Sept 2006, pp. 293–310.

[12] Furon, T., F. Cayre, and C. Fontaine, "Watermarking security: theory and practice," *Signal Processing, IEEE Transactions on*, Institute of Electrical and Electronics Engineers (IEEE), Vol. 53, No. 10, 2005, pp. 3976–3987.

[13] Zhang, D., J. Ni, and D.-J. Lee, "Security Analysis for Spread-Spectrum Watermarking Incorporating Statistics of Natural Images," in *Advances in Visual Computing*, Vol. 5359 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 400–409, 2008.

[14] Shannon, C., "Communication theory of secrecy systems," *Bell System Technical Journal*, Vol. 28, October 1949, pp. 656–715.

[15] Pérez-Freire, L., and F. Pérez-González, "Spread-Spectrum Watermarking Security," *Information Forensics and Security, IEEE Transactions on*, Vol. 4, No. 1, March 2009, pp. 2–24.

[16] Pérez-Freire, L., et al., "Security of lattice-based data hiding against the Known Message Attack," *Information Forensics and Security, IEEE Transactions on*, Vol. 1, No. 4, December 2006, pp. 421–439.

[17] Pérez-Freire, L., and F. Pérez-González, "Security of Lattice-Based Data Hiding Against the Watermarked-Only Attack," *Information Forensics and Security, IEEE Transactions on*, Vol. 3, No. 4, Dec 2008, pp. 593–610.

[18] Cox, I., G. Doërr, and T. Furon, "Watermarking is not cryptography," in *Proceedings of the International Workshop on Digital Watermarking*, Vol. 4283 of *Lecture Notes in Computer Science*, Jeju Island, Korea, 2006, pp. 1–15.

[19] Kalker, T., "Considerations on watermarking security," in *Proceedings of the Fourth Workshop on Multimedia Signal Processing (MMSP)*, IEEE, Cannes, France, October 2001, pp. 201–206.

[20] Bas, P., and T. Furon, "A New Measure of Watermarking Security: The Effective Key Length," *Information Forensics and Security, IEEE Transactions on*, Institute of Electrical and Electronics Engineers (IEEE), Vol. 8, No. 8, Jul. 2013, pp. 1306 – 1317.

[21] Bas, P., and T. Furon, "Key length Estimation of zero-bit watermarking schemes," in *EUSIPCO - 20th European Signal Processing Conference*, Romania, Aug. 2012, p. TBA.

[22] Furon, T., and P. Bas, "A New Measure of Watermarking Security Applied on DC-DM QIM," in *IH - Information Hiding*, Berkeley, United States, May 2012, p. TBA.

[23] Katzenbeisser, S., "Computational security models for digital watermarks," in *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Montreux, Switzerland, April 2005, pp. 1261–1282.

[24] Doërr, G., and J. Dugelay, "Security pitfalls of frame-by-frame approaches to video watermarking," *Signal Processing, IEEE Transactions on*, Vol. 52, No. 10, Oct 2004, pp. 2955–2964.

[25] Bas, P., and A. Westfeld, "Two Key Estimation Techniques for the Broken-Arrows Watermarking Scheme," in *ACM Multimedia and Security Workshop 2009*, Princeton, NJ, United States, 2009, pp. 1–8.

[26] Bas, P., and G. Doërr, "Practical Security Analysis of Dirty Paper Trellis Watermarking," in *Information Hiding*, Vol. 4567 of *Lecture Notes in Computer Science*, Springer, June 2007, pp. 174–188.

[27] Bas, P., and G. Doërr, "Evaluation of an Optimal Watermark Tampering Attack Against Dirty Paper Trellis Schemes," in *ACM Multimedia and Security Workshop 2008*, United Kingdom, 2008, pp. 227–232.

[28] Valizadeh, A., and J. Wang, "Correlation-and-Bit-Aware Spread Spectrum Embedding for Data Hiding," *Information Forensics and Security, IEEE Transactions on*, Vol. 6, No. 2, June 2011, pp. 267–282.

[29] Linnartz, J.-P., and M. van Dijk, "Analysis of the Sensitivity Attack against Electronic Watermarks in Images," in *Information Hiding*, Vol. 1525 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 258–272, 1998.

[30] Cox, I. J., and J.-P. Linnartz, "Public watermarks and resistance to tampering," in *Proceedings of the International Conference on Image Processing, ICIP'97*, Vol. 3, Oct 1997, pp. 3–6.

[31] Comesaña, P., L. Pérez-Freire, and F. Pérez-González, "Blind Newton Sensitivity Attack," *IEE Proc. on Information Security*, Vol. 153, No. 3, 2006, pp. 115–125.

[32] Comesaña, P., *Side-informed data hiding: robustness and security analysis*, Ph.D. thesis, Universidade de Vigo, 2006.

[33] Earl, J., "Tangential sensitivity analysis of watermarks using prior information," in *Security, steganography and watermarking of multimedia contents IX*, Vol. 6505 of *Proceedings of SPIE-IS&T Electronic Imaging*, 2007.

[34] Furon, T., G. Silvestre, and N. Hurley, "JANIS: Just Another N-order side-Informed Scheme," in *Proceedings of the International Conference on Image Processing, ICIP'02*, Vol. 2, Rochester, NY, USA, September 2002, pp. 153–156.

[35] Kalker, T., J.-P. Linnartz, and M. van Dijk, "Watermark estimation through detector analysis," in *Proceedings of the International Conference on Image Processing, ICIP'98*, Vol. 1, Oct 1998, pp. 425–429.

[36] El Choubassi, M., and P. Moulin, "Noniterative Algorithms for Sensitivity Analysis Attacks," *Information Forensics and Security, IEEE Transactions on*, Vol. 2, No. 2, June 2007, pp. 113–126.

[37] El Choubassi, M., and P. Moulin, "On Reliability and Security of Randomized Detectors Against Sensitivity Analysis Attacks," *Information Forensics and Security, IEEE Transactions on*, Vol. 4, No. 3, Sept 2009, pp. 273–283.

[38] Mansour, M., and A. Tewfik, "Secure detection of public watermarks with fractal decision boundaries," in *Signal Processing Conference, 2002 11th European*, Sept 2002, pp. 1–4.

[39] Furon, T., and P. Bas, "Broken Arrows," *EURASIP Journal on Information Security*, Hindawi, Vol. 2008, Oct. 2008, pp. ID 597040.

[40] Westfeld, A., "Fast Determination of Sensitivity in the Presence of Countermeasures in BOWS-2," in *Information Hiding*, Vol. 5806 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 89–101, 2009.

[41] Choubassi, M. E., and P. Moulin, "On the fundamental tradeoff between watermark detection performance and robustness against sensitivity analysis attacks," in *Security, steganography, and watermarking of multimedia content VIII*, Vol. 6072 of *Proceedings of SPIE-IS&T Electronic Imaging*, SPIE, San Jose, CA, USA, 2006, p. 1I.

[42] Comesaña, P., and F. Pérez-González, "Breaking the BOWS Watermarking System: Key Guessing and Sensitivity Attacks," *EURASIP Journal on Information Security*, Vol. 2007, No. 2, February 2007. Article ID 25308.

[43] Barni, M., et al., "Are you threatening me? Towards smart detectors in watermarking," in *Proc. of SPIE Media Watermarking, Security, and Forensics*, San Francisco, CA, USA, Feb. 2014.

[44] Kutter, M., S. Voloshynovskiy, and A. Herrigel, "Watermark copy attack," in *Proc. SPIE Security and Watermarking of Multimedia Contents II*, Vol. 3971, 2000, pp. 371–380.