Chapter 2
A Quick Tour of Watermarking Techniques

In order to understand and analyse the main components of watermarking security presented in the next chapters, we introduce in this chapter the different elements needed to embed a watermark or a message inside a host content. We first present a functional view of a watermarking scheme (the embedding function, the decoding/detection function) and then its geometrical interpretation. Then we present the most popular class of watermarking schemes: spread-spectrum watermarking and watermarking techniques based on the idea of dirty paper codes.

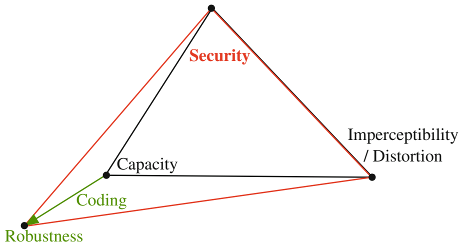
2.1 Basic Principles

2.1.1 Watermarking Constraints

As presented in Fig. (1.3), the three main constraints of watermarking are security, imperceptibility and capacity. Imperceptibility measures the distortion between host contents and watermarked contents, it can be measured using classical l-norm distances or perceptive measures derived from the human system. Capacity measures the amount of information that can be transmitted by the watermarking system through the channel, it can be computed using information theoretical measures such as the mutual information. Robustness can be seen as a practical translation of capacity and can be measured by the symbol error rate through the watermarking channel. In order to have a practical watermarking system whose performances are close to capacity, a system has to use an appropriate channel coding system. The last constraint, security is more complex and defined deeply within this book, particularly in Chap. 3. Figure 2.1 illustrates two triangles of constraints, a theoretical one in black takes into account capacity, and a practical one in red takes into account robustness.

P. Bas et al., *Watermarking Security*, SpringerBriefs in Signal Processing,
DOI 10.1007/978-981-10-0506-0_2

Fig. 2.1 The different constraints of Watermarking



2.1.2 Zero-Bit and Multi-bit Watermarking

Watermarking schemes are divided into two classes called zero-bit and multi-bit watermarking.

For zero-bit watermarking, a watermark is embedded in a host content and the goal is to detect if a content is watermarked or not. Watermark detection can therefore be seen as an hypothesis test with two hypotheses: H_0 : “content no watermarked” and H_1 : “content watermarked”. The performance of a zero-bit watermarking scheme is evaluated regarding the false alarm probability or false detection probability p_{fd} (i.e. the probability to detect a watermark in a non-watermarked content) and the miss detection probability (i.e. the probability to not detect a watermark in a watermarked content).

For multi-bit watermarking, a message m is embedded in a host content and the goal is to decode the embedded message. The performance of a multi-bit watermarking scheme is evaluated with respect to the message error rate, i.e. probability to wrongly decode a message, or the bit error rate, i.e. the probability to decode wrongly embedded bits.

We now present two equivalent formalisations of watermarking, the first one is the processing view, where both the embedding and the decoding schemes are seen as processes or functions. The second view is more geometrical and allows us to picture effects of embedding, decoding and also the role of the secret key in Watermarking.

2.1.3 A Processing View

The main chain of processes is the following: from a content $\mathbf{c}_\mathbf{x}$ is extracted an host signal $\mathbf{x} \in \mathcal{X}$ which is transformed into a watermarked content $\mathbf{y} \in \mathcal{X}$ using a secret key $\mathbf{k} \in \mathcal{K}$.

The extraction process $\psi(.)$ is public (it does not depend of a secret key) and invertible, $\mathbf{x} = \psi(\mathbf{c}_\mathbf{x})$. It can be for example the Least Significant Bits of an image, its Discrete Cosine Transform coefficients or Wavelets subbands. An inverse function $\psi^{-1}(.)$ is used to generate the watermarked content $\mathbf{c}_\mathbf{y} = {}^{\mathbf{-1}}(\mathbf{y}, \overline{\mathbf{x}})$, where $\overline{\mathbf{x}}$ represents the components of $\mathbf{c}_\mathbf{x}$ which are not present in $\mathbf{x}$. Because watermarking processes can be used on any signal $\mathbf{x}$, input signal $\mathbf{c_x}$ and output signal $\mathbf{c_y}$ will be omitted in the rest of this chapter.

A message m is embedded, if m is a power of 2, it can be decomposed into a set of N_m bits $\{b_1, \ldots, b_{N_m}\}$.

The embedding is performed using an embedding function $e(.)$ to generate the watermarked content:

$$\mathbf{y} = e(\mathbf{x}, \mathbf{k}, m), \tag{2.1.1}$$

where $\mathbf{k}$ denotes the secret key. It enables for example to select a subset of coefficients of $\mathbf{c_x}$, or to generate a set of pseudo random vectors seeded with $\mathbf{k}$ used to build a secret subspace or to scramble the content space $\mathcal{X}$. This embedding can be seen as an additive process involving the watermark vector $\mathbf{w} = \mathbf{y} - \mathbf{x}$.

The decoded message is estimated using a decoding function

$$\hat{m} = d(\mathbf{z}, \mathbf{k}), \tag{2.1.2}$$

from a potentially corrupted vector

$$\mathbf{z} = \mathbf{y} + \mathbf{n}, \tag{2.1.3}$$

where $\mathbf{n}$ is an additive noise. We can unify multi-bit and zero-bit schemes by embedding a constant message $m = \text{cst} = 1$ for zero-bit watermarking. The decoding function consequently becomes a detection function: $\hat{m} = d(\mathbf{z}, \mathbf{k}) = 1$ if the watermark is detected and $\hat{m} = d(\mathbf{z}, \mathbf{k}) = 0$ if not.

In this chain of processes, we can highlight the watermarking channel which converts the message m into a watermarked content $\mathbf{y}$ and decodes the message $\hat{m}$ from the corrupted watermarked content $\mathbf{z}$. This channel is private and accessible using the secret key $\mathbf{k}$ or as we will see in Sect. 3.3 perturbed versions of $\mathbf{k}$. The embedding and decoding functions are considered as public (this is the Kerckhoffs' principle which states that the security relies only on the usage of a secret key) and one goal of the adversary is to have access to the watermarking channel. By doing so, the adversary is able to embed a new message, to alter the current one, to copy it into another content, or to decode it from a watermarked content. The access to the watermarking channel can be total (a read and write access) or only partial (the possibility to copy the message into another content, or to alter it).

2.1.4 The Geometrical View

A more geometrical view of the effects of embedding and decoding in the content space $\mathcal{X}$ is now presented. This view is motivated by the decoding function $d(.)$ which induces two entities:

Table 2.1 Equivalences between processing and the geometrical views

Processing view	Geometrical view
Embedding function $e(.)$	Choice of the watermark vector $\mathbf{w}$
Embedded message m	Subset of $\mathcal{D}$ having label m
Decoding function $d(.)$	Identification of the label $\mathrm{m}(i)$ such that $\mathbf{z} \in \mathcal{D}_i$
Secret key $\mathbf{k}$	Set of decoding regions $\mathcal{D}$ and the labelling function $\mathrm{m}(.)$

1. a partition of $\mathcal{X}$ into a set of N_d decoding regions $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_{N_d}\}$, N_d can be infinite,
2. a labelling function $\mathrm{m}(.)$ which is a surjective function from $[1 \mathinner{..} N_d]$ to $[1 \mathinner{..} N_m]$ (hence $N_d \geq N_m$).

Each decoding region $\mathcal{D}_i$ is labeled by a message $m_i = \mathrm{m}(i)$ and the function $d(.)$ assigns to $\mathbf{z}$ the label of the decoding region it belongs to:

$$\hat{m} = \mathrm{m}(i) \text{ if } \mathbf{z} \in \mathcal{D}_i. \tag{2.1.4}$$

Note that in the case of zero-bit watermarking, there is only two possible labels $m(i) = 0$ or $m(i) = 1$. In this case, the content $\mathbf{z}$ is detected as watermarked only if $\hat{m} = 1$ and $\hat{m} = 0$ means that the content is detected as original.

The set $\mathcal{D}$ can be generated from the Voronoï cells of N_d points of $\mathcal{X}$ (see for example Sects. 2.4 and 2.4.3) or by the boundaries of the decoding regions (see for example Sects. 2.2 and 2.3). In order to minimise the embedding distortion, the embedding function $e(.)$ in going to "push" the host content $\mathbf{x}$ into the closest decoding region $\mathcal{D}_j$ labeled with the message m to embed (i.e. satisfying $m = \mathrm{m}(j)$) by adding a properly chosen watermark vector $\mathbf{w}$ to obtain the watermarked content $\mathbf{y} = e(\mathbf{x}, \mathbf{k}, m) = \mathbf{x} + \mathbf{w}$. The equivalence between the elements of the processing view and those of the geometrical view are summed up in Table 2.1 (Fig. 2.2).

2.1.5 Three Fundamental Constraints

A watermarking scheme has also to take into account three fundamental constraints presented below.

The **Robustness**, which can be defined as the probability to correctly decode the embedded message from $\mathbf{z}$. Usually, the probability to wrongly decode the message is used instead and is computed using the message error rate $p_{me} = \mathrm{E}_M[\mathbb{P}(\hat{M} \neq M)]$ or the bit error rate $p_{be} = \mathrm{E}_B[\mathbb{P}(\hat{B} \neq B)]$ when the message is composed of a set of embedded bits. The goal of the watermarking designer will be to maximise the robustness, i.e. to minimise the probability of error. In order to do so, the embedding function will try to push the watermarked content inside the appropriate decoding region. The farther $\mathbf{y}$ is from the boundaries of $\mathcal{D}_i$ the more important is the robustness.

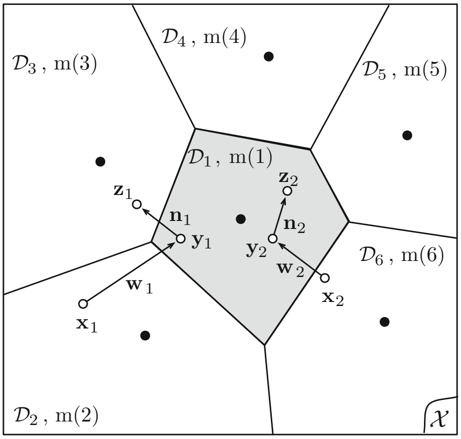


Fig. 2.2 A geometrical view of watermarking: in this example host vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ are used to embed the same arbitrary message $m = \mathrm{m}(1)$ and consequently have to be moved in region $\mathcal{D}_1$. Symbol • represents centres of the Voronoï cells defining the decoding regions, and ∘ represents contents in the content space $\mathcal{X}$. An embedding function generates two watermark vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ which are added to create the watermarked contents $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively. In this example, the first embedding is less robust than the second one because $\mathbf{y}_1$ is closer to the boundary of $\mathcal{D}_1$ than $\mathbf{y}_2$. The message error rate will be consequently more important for the first content than for the second one (for a same Watermark to Noise Ratio under the AWGN channel). The noise $\mathbf{n}_1$ generates a decoding error but not the noise $\mathbf{n}_2$

The probability of decoding error is usually computed for a given Watermark to Noise Ratio, defined as the WNR $= \log_{10}\left(\mathrm{E}[W^2]/\mathrm{E}[N^2]\right)$ dB. Note also that another important and popular constraint, the **Capacity**, can be seen as the dual constraint of Robustness: for a given channel, the watermarking designer wants either to reach the capacity in order to transmit as much information as possible; or for a given transmission rate, he wants to maximise robustness, i.e. to minimise the message error rate. In each case however the watermarking designer will have to find the best code given his objective, the code maximising robustness for a given channel will be the same than the code maximising the transmission rate, i.e. reaching the capacity.

The **Distortion**, classically defined as the power of the watermark signal $\mathrm{E}[W^2]$ or the Document to Watermark Ratio DWR $= \log_{10}\left(\mathrm{E}[X^2]/\mathrm{E}[W^2]\right)$ dB and is equal to DWR $= 10 \cdot \log_{10}\left(\sigma_x^2/\sigma_w^2\right)$ dB if both signals are centred. Note that the distortion measure can be more elaborated in order to take into account psychovisual criterions. The goal of the watermarking designer will be to minimise the distortion or to set it below a threshold. This means that the distance between $\mathbf{x}$ and $\mathbf{y}$, or its expectation, has to be bounded or minimised. The same rule applies for the distance between between $\mathbf{x}$ and the different decoding regions.

The **Security**, defined by Kalker as the inability for the adversary to have access to the watermarking channel [1], which can be translated as the inability to estimate the secret key $\mathbf{k}$ (or an approximation of it) granting the reading or writing access of the message. Under the geometrical viewpoint, **k** allows us to generate the set $\mathcal{D}$ and the labelling function m(.). We can see that if the adversary knowns these two entities, he is able to have access to the watermarking channel, e.g. he can embed or decode messages. This last constraint will motivate the next chapters of this document and will be formalised in the next chapter.

Note that the key **k** which will be used in the following of the document is generated from a user key $\mathbf{k}_u$ which is defined by the user who wants to embed of decode the message. The key which is used to have access to the watermarking channel in the content space $\mathcal{X}$, it not $\mathbf{k}_u$ but **k** which represents a set of signals and parameters that are generated from $\mathbf{k}_u$. From a security point of view, finding the user key $\mathbf{k}_u$ is equivalent to finding **k** or the set $\{\mathcal{D}, \mathrm{m}(.)\}$: each entity allows us to have a complete access to the watermarking channel. We consequently have the following relation $\mathbf{k}_u \rightarrow \mathbf{k} \rightarrow \{\mathcal{D}, \mathrm{m}(.)\}$. In the sequel, we will omit to define the user key since it is related to the implementation of the watermarking system, and it classically corresponds to the seed of a pseudo-random generator.

It is also important to note that these three constraints are strongly dependant from each other. For example, the robustness grows when the decoding regions become large and the contents goes deep inside one of them; however such a strategy implies that the distortion grows as well. A balance between distortion and robustness has consequently to be found. We will see in the next chapter that the same applies regarding the security.

We now present a quick tour of the most popular watermarking schemes and we detail the principles of embedding, decoding or detection functions.

2.2 Spread Spectrum and Improved Spread Spectrum

Spread Spectrum Watermarking

Also abbreviated SS, it uses the same principles than Spread Spectrum communications to transmit a message in a noisy environment [2]. The bandwidth of the message is spread on a larger bandwidth thanks to a modulation with one or several carriers represented by pseudo-random vectors.

For this basic implementation, SS watermarking allows us to embed $N_b = \log_2(N_m)$ bits $(b_1, \ldots, b_{N_b})$. Under given security scenarios, security is granted thanks to the use of a set of N_m pseudo-random vectors $\mathbf{k} = \{\mathbf{k}_1, \ldots, \mathbf{k}_{N_m}\}$ seeded by the user key (see Sect. 2.1.5). Without loss of generality, we can normalise and orthogonalize each secret vector $(\forall (i \neq j), ||\mathbf{k}_i|| = 1$ and $\mathbf{k}_i^t \mathbf{k}_j = 0)$, and the embedding formula is:

$$\mathbf{y} = e(\mathbf{x}, m, \mathbf{k}) = \mathbf{x} + \alpha \sum_{i=1}^{N_b} (-1)^{b_i} \mathbf{k}_i. \tag{2.2.1}$$

The watermark vector is $\mathbf{w} = \alpha \sum_{i=1}^{N_b} (-1)^{b_i} \mathbf{k}_i$, and $||\mathbf{w}||^2 = N_b \alpha^2 = \text{cst}$. Note that α is a scalar used to choose the embedding distortion $\text{DWR} = 10 \cdot \log_{10}\left(\frac{N_v \sigma_x^2}{N_b \alpha^2}\right)$ dB which means that

$$\alpha = \sqrt{N_v/N_b}\sigma_x 10^{-\text{DWR}/20}. \tag{2.2.2}$$

The decoding is performed on the potentially corrupted vector $\mathbf{z}$ by computing $\mathbf{z}^T\mathbf{k}_i$, i.e. the projections of the watermarked vector on each pseudo-random vector:

$$\hat{b}_i = 2 \cdot \text{sign}(\mathbf{z}^T\mathbf{k}_i) - 1. \tag{2.2.3}$$

The boundaries of the decoding regions are consequently defined by the N_b hyper plans orthogonal to each $\mathbf{k}_i$ which yields to 2^{N_b} different decoding regions, (one for each message). Figure 2.3 represents the decoding regions and the locations of watermarked contents projected on the plan $(O, \mathbf{k}_1, \mathbf{k}_2)$ for $N_b = 2$ and $(O, \mathbf{k}_1, \text{u}_\perp)$ for $N_b = 1$ where $\mathbf{u}_\perp$ is a random unitary vector orthogonal to $\mathbf{k}_1$. Under an AWGN channel defined by a variance σ_n^2 and $\text{WNR} = 10\log_{10}\left(\alpha^2/\sigma_n^2\right)$, the BER is computed using the projection $\boldsymbol{Z}^T\boldsymbol{K}_i$ which is a Gaussian random variable of law $\mathcal{N}(\alpha, \sigma_x^2 + \sigma_n^2)$, and the BER is given by:

$$p_{be} = \frac{1}{2}\left[1 + \text{erf}\left(\frac{-\alpha}{\sqrt{2\left(\sigma_x^2 + \sigma_n^2\right)}}\right)\right] = \frac{1}{2}\left[1 + \text{erf}\left(-\sqrt{\frac{N_v 10^{-\text{DWR}/10}}{2N_b\left(1 + 10^{-(\text{WNR}+\text{DWR})/10}\right)}}\right)\right]. \tag{2.2.4}$$

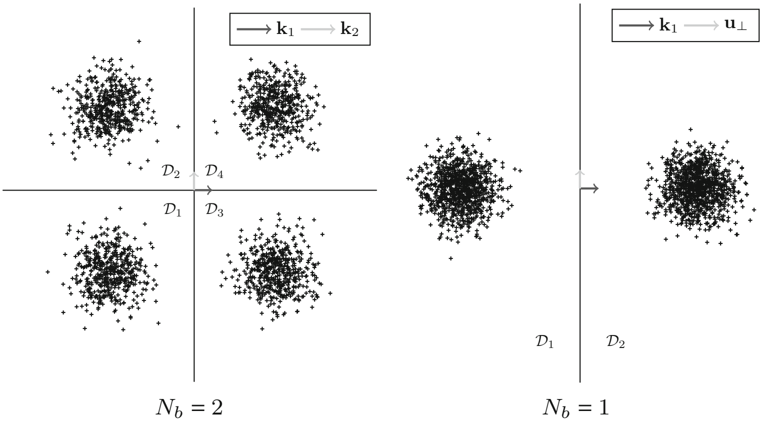


Fig. 2.3 Geometrical view of SS watermarking—Watermarked contents projected on $\mathbf{k}_1$ and $\mathbf{k}_2$ ($N_b = 2$) or $\mathbf{u}_\perp$ ($N_b = 1$) and the associated decoding regions. $N_v = 128$, Gaussian host, $\text{DWR} = 5$ dB

2.2.1 Improved Spread Spectrum

Improved Spread Spectrum (ISS) watermarking [3] is a variant of Spread Spectrum and is motivated by the fact that for SS the embedding (2.2.1) uses only one parameter α. All host contents are consequently moved by a watermark vector $\mathbf{w}$ of constant norm, only the direction depends on the message to embed. In order to take into account the robustness constraint, the authors of [3] have proposed to use a second parameter γ which is tuned in order to minimise the bit error rate for a given WNR under an AWGN channel. The term $\gamma \in [0, 1]$ is chosen to perform **host rejection**, which consists in reducing the interference (or intercorrelation) $\mathbf{x}^T\mathbf{k}_i$ between the host $\mathbf{x}$ and each pseudo-random vector $\mathbf{k}_i$ by adding the term $-\gamma\left(\mathbf{x}^T\mathbf{k}_i\right)\mathbf{k}_i$, and it is equivalent to reducing the variance of the watermarked contents along each $\mathbf{k}_i$. Once the variance is reduced, the content is moved using a constant vector, proportional to $\mathbf{k}_i$ using a parameter α similar to the one used in Spread Spectrum:

$$\mathbf{y} = e(\mathbf{x}, m, \mathbf{k}) = \mathbf{x} + \sum_{i=1}^{N_b} \left(\alpha(-1)^{m_i} - \gamma \mathbf{x}^T\mathbf{k}_i\right)\mathbf{k}_i. \tag{2.2.5}$$

The watermark vector is consequently $\mathbf{w} = \sum_{i=1}^{N_b}\left(\alpha(-1)^{m_i} - \gamma\mathbf{x}^T\mathbf{k}_i\right)\mathbf{k}_i$ and $||\mathbf{w}||^2 = N_b\left(\alpha^2 + \gamma^2\sigma_x^2\right)$ which gives

$$\text{DWR} = 10 \cdot \log_{10}\left(\frac{N_v\sigma_x^2}{N_b\left(\alpha^2 + \gamma^2\sigma_x^2\right)}\right)\text{dB} \tag{2.2.6}$$

For a given γ this means that we have:

$$\alpha = \sqrt{\left(\frac{N_v\sigma_x^2}{N_b}\right)10^{-\text{DWR}/10} - \sigma_x^2\gamma^2} \text{ if } \left(\frac{N_v\sigma_x^2}{N_b}\right)10^{-\text{DWR}/10} \geq \sigma_x^2\gamma^2. \tag{2.2.7}$$

Note that for $\gamma = 0$, SS and ISS are identical. Figure 2.4 depicts the locations of watermarked contents using ISS embedding for $\gamma = 0.9$. We can notice that the host rejection strategy allows us to reduce the variance of the watermarked contents inside each decoding region, making them statistically less prone to leave the decoding region for a noise of relative small power than contents watermarked using SS.

The decoding of the message is performed exactly as for SS, i.e. using Eq. (2.2.3). The authors propose to compute γ in order to minimise the BER for an AWGN channel of given WNR. Under the AWGN, the projection $\boldsymbol{Z}^T\boldsymbol{K}_i$ is a Gaussian random variable of law $\mathcal{N}(\alpha, (1-\gamma)^2\sigma_x^2 + \sigma_n^2)$ and the BER is given by:

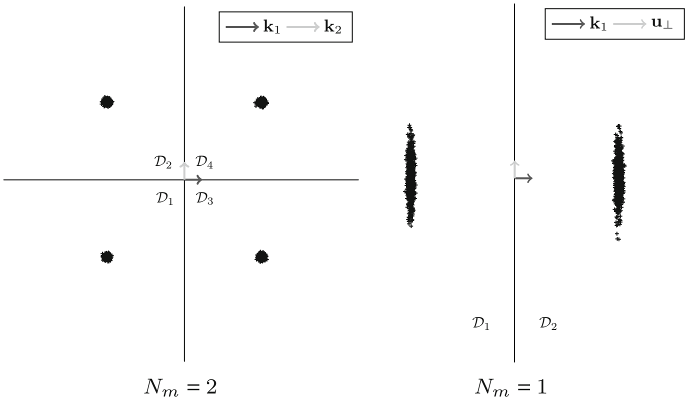


Fig. 2.4 Geometrical view of ISS watermarking—Watermarked contents projected on $\mathbf{k}_1$ and $\mathbf{k}_2$ ($N_b = 2$) or $\mathbf{u}_\perp$ ($N_b = 1$) and the associated decoding regions. $N_v = 128$, Gaussian host, $\text{DWR} = 5\,\text{dB}$, $\gamma = 0.9$

$$\begin{aligned} p_{be} &= \frac{1}{2}\left[1+\operatorname{erf}\left(-\sqrt{\frac{\left(\frac{N_v\sigma_x^2}{N_b}\right)10^{-\frac{\text{DWR}}{10}}-\sigma_x^2\gamma^2}{2\left((1-\gamma)^2\sigma_x^2+\sigma_n^2\right)}}\right)\right],\\ &= \frac{1}{2}\left[1+\operatorname{erf}\left(-\sqrt{\frac{1}{2}\frac{\left(\frac{N_v}{N_b}\right)10^{-\frac{\text{DWR}}{10}}-\gamma^2}{(1-\gamma)^2+10^{-\frac{\text{WNR}+\text{DWR}}{10}}}}\right)\right]. \end{aligned} \tag{2.2.8}$$

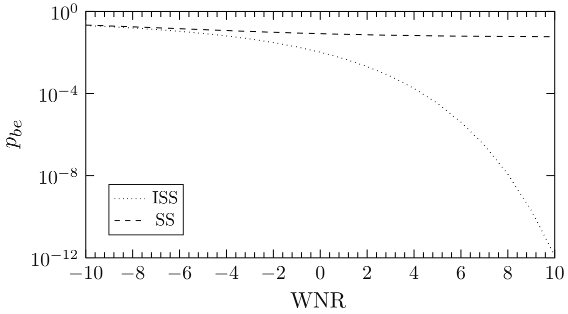
The value $\gamma^{(R)}$ which minimises BER is given by solving $\partial p_{be}/\partial\gamma = 0$ and gives

$$\gamma^{(R)} = \frac{1}{2}\left((1+b+a) - \sqrt{(1+b+a)^2 - 4a}\right), \tag{2.2.9}$$

with $a = (N_v/N_b)\cdot 10^{-\text{DWR}/10}$ and $b = \sigma_n^2/\sigma_x^2 = 10^{-(\text{WNR}+\text{DWR})/10}$.

The comparison between the two BERs is illustrated on Fig. 2.5, we can notice that when the noise power increases, the performance of ISS embedding tends to be the

Fig. 2.5 Comparison between the BER for ISS and SS w.r.t. the WNR ($N_v = 64$, $N_b = 8$, $\text{DWR} = 5\,\text{dB}$)



same than the performance of SS embedding. This is due to the fact that $\gamma^{(R)} \to 0$ when WNR $\to -\infty$. On the other hand, the improvement of ISS regarding SS is really important for low WNRs, i.e. when the host rejection is important and $\gamma^{(R)} \to 1$.

2.3 Correlation Based Zero-Bit Watermarking

The principle of spread spectrum watermarking can also be used for zero-bit watermarking. As for SS decoding, the detection is also performed using the correlation $\mathbf{z}^T\mathbf{k}$ between the secret key $\mathbf{k}$ and the watermarked (and possibly corrupted) content $\mathbf{z}$. However, since the watermark has to be detected also on possibly scaled versions of the watermarked contents $\mathbf{z} = \alpha\mathbf{y}$, the detection is performed by computing the normalised correlation instead:

$$\begin{aligned} &d(\mathbf{z}, \mathbf{k}) = 1 \quad \text{if} \quad \tfrac{|\mathbf{z}^T\mathbf{k}|}{||\mathbf{z}^T|| \cdot ||\mathbf{k}||} > T, \\ &d(\mathbf{z}, \mathbf{k}) = 0 \text{ else.} \end{aligned} \tag{2.3.1}$$

Note that $|\mathbf{z}^T\mathbf{k}| / \left(||\mathbf{z}^T|| \cdot ||\mathbf{k}||\right) = |\cos\theta|$ where θ is defined as the angle between the two vectors $\mathbf{k}$ and $\mathbf{z}$. This means that the boundary of the detection region is defined by a double hyper-cone (called here $\mathcal{C}$) of axis $\mathbf{k}$ and angle $\theta = \arccos(T)$. The threshold T (or θ) is computed in order to satisfy a given false alarm probability $\mathbb{P}[d(\boldsymbol{X}, \boldsymbol{K}) = 1] = p_{fa}$. Since $\boldsymbol{K}$ is uniformly distributed on an hyper sphere of dimension N_v, this probability can be computed as the ratio between twice the surface of the hyper-cap of solid angle θ over the surface of the hyper-sphere [4]:

$$p_{fa} = 1 - I_{\cos^2\theta}\left(1/2, (N_v - 1)/2\right), \tag{2.3.2}$$

where $I(.)$ denotes the regularised incomplete beta function.

Without loss of generality we set $||\mathbf{k}|| = 1$. Once the decoding region is defined, the embedding has to be designed in order to move the host content $\mathbf{x}$ into $\mathcal{C}$. In order to find the adequate subspace to choose $\mathbf{w}$, [5] proposed to build a plan $(O, \mathbf{k}, \mathbf{e}_2)$ defined by the secret key $\mathbf{k}$ (the cones axis) and the unitary vector $\mathbf{e}_2$ orthogonal to $\mathbf{k}$, such that the host vector belongs to the plan $(O, \mathbf{k}, \mathbf{e}_2)$:

$$\mathbf{e}_2 = \frac{\mathbf{x} - \left(\mathbf{x}^T\mathbf{k}\right)\mathbf{k}}{||\mathbf{x} - \left(\mathbf{x}^T\mathbf{k}\right)\mathbf{k}||}. \tag{2.3.3}$$

Since the closest point to $\mathbf{x}$ included in $\mathcal{C}$ belongs to $(O, \mathbf{k}, \mathbf{e}_2)$, this plan spans the shortest direction to enter inside the cone.

Different strategies are possible in order to choose the watermark vector $\mathbf{w}$. For all these strategies, we assume that the norm of the watermark vector is constant and equal to $D = \sqrt{N_v\sigma_x^2 \cdot 10^{-\mathrm{DWR}/10}}$.

In [5], the authors propose to assume that the noise $\mathbf{n}$ and the watermarked content $\mathbf{y}$ are orthogonal ($\mathbf{y}^T\mathbf{n} = 0$), this assumption is more and more realistic when N_v grows and the content suffers an AWGN channel. The goal here is to find the point inside $\mathcal{C}$ such that the distance R between $\mathbf{y}$ and the cone boundary considering a direction orthogonal $\mathbf{y}$ is maximal. Thanks to the Pythagorus theorem,

$$R = \sqrt{(\mathbf{y}[1]\tan\theta)^2 - \mathbf{y}[2]^2}, \tag{2.3.4}$$

and we can use a basic optimisation technique to find $\mathbf{y}_{\text{ON}}$ such that R is maximal (ON stands for Orthogonal Noise). See Fig. 2.6 for a geometrical illustration.

In [6] (see also Sect. 4.3 of this book), the authors propose to maximise the robustness considering the worst case scenario, i.e. the fact that the adversary knows the cone axis. In this setup, the robustness can be defined as the distance between $\mathbf{y}$ and the nearest point of the cone boundary. The solution to maximise this distance is to move $\mathbf{x}$ in a direction Orthogonal to the Boundary (hence the denomination OB). In certain cases the watermarked content can reach the cone axis, and because of the axial symmetry of the cone, the best strategy then is to go in the same direction than $\mathbf{k}$ once the axis is reached. The embedding formulae is literal and can be written as:

$$\text{if } D < \frac{\mathbf{x}[2]}{\cos\theta} : \begin{cases} \mathbf{y}_{\text{OB}}[1] = & \mathbf{x}[1] + D \cdot \sin\theta \\ \mathbf{y}_{\text{OB}}[2] = & \mathbf{x}[2] - D \cdot \cos\theta \end{cases} \tag{2.3.5}$$

and

$$\text{if } D \geq \frac{\mathbf{x}[2]}{\cos\theta} : \begin{cases} \mathbf{y}_{\text{OB}}[1] = & \mathbf{x}[1] + \sqrt{D^2 - \mathbf{x}[2]^2} \\ \mathbf{y}_{\text{OB}}[2] = & 0 \end{cases}. \tag{2.3.6}$$

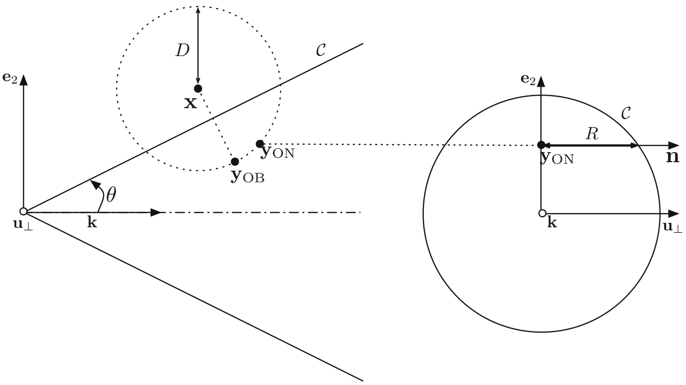


Fig. 2.6 *Left* geometrical presentations of the 3 different embedding strategies, Orthogonal Noise ($\mathbf{y}_{\text{ON}}$), Orthogonal to the Boundary ($\mathbf{y}_{\text{OB}}$) and by Maximizing the Mutual Information ($\mathbf{y}_{\text{MMI}}$). *Right* computation of the robustness R when assuming that the noise is orthogonal to the watermarked content

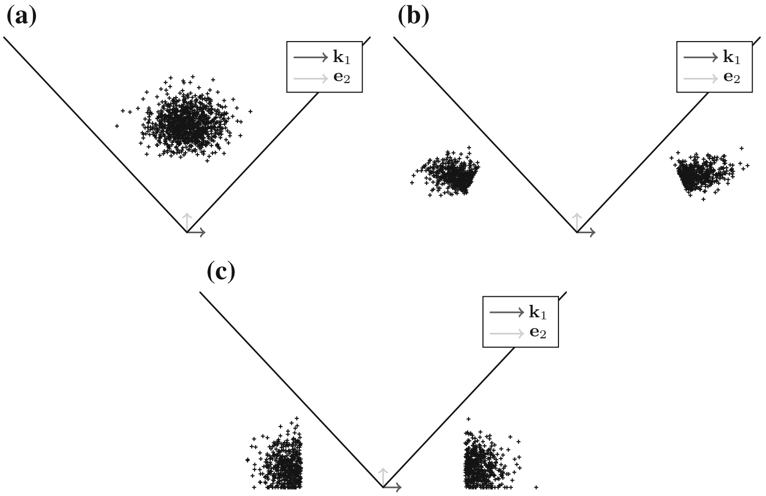


Fig. 2.7 Projection of the original contents, **a** and watermarked contents **b** orthogonal noise, **c** orthogonal to the boundary and **d** Max Mutual Information in the $(O, \mathbf{k}, \mathbf{e}_2)$. $N_v = 32$, DWR $= 0$ dB, $\theta = \pi/4$ ($p_{fa} = 4 \cdot 10^{-6}$)

The geometric illustration of the embedding "Orthogonal to the Boundary" is depicted on Fig. 2.6.

Figure 2.6 represents the geometrical interpretations of the two different embeddings in the plan $(O, \mathbf{k}, \mathbf{e}_2)$ (left) and in the plan plan $(O, \mathbf{u}_\perp, \mathbf{e}_2)$ for the first strategy (right).

Figure 2.7 show the projection of original and watermarked contents for the 3 different embedding, we can notice that only the second strategy brings few contents to the cone axis.

2.4 Watermarking Based on Dirty Paper Codes

If host rejection is one way to increase the robustness of the watermarking scheme by reducing the variance of the watermarked content inside the decoding regions, another strategy consists in generating several decoding regions for the same message. Such a strategy is inspired by the idea of "writing on dirty papers" proposed by Costa [7] who shows theoretically that for an AWGN channel, the capacity of the watermarking system solely depends on the variance of the noise and the embedding distortion, and is not dependant of the host signal (e.g. the "dirty paper").

The practical construction of such a "dirty paper" scheme relies on the generation of M bins, each bin containing codewords. For a host content $\mathbf{x}$ and a message to embed m, the embedding distortion is minimised by considering only the mth bin and going toward the nearest codeword of this bin. Such a strategy is also called *informed coding* [8] since the embedding is informed by taking into account the host state $\mathbf{x}$ and picking the right codeword that codes the message to embed m.

Under the geometric view, it means that if the host vector $\mathbf{x}$ is too far for one embedding region, the strategy proposed by Costa's paper is to select another embedding region labeled by the same message but closer to $\mathbf{x}$, the number of embedding regions being limited by the robustness constraint.

Practically, the implementation of dirty paper watermarking implies several requirements:

- a function to generate the N_m bins in $\mathcal{X}$, which corresponds to the labelling function defined in Sect. 2.1.4,
- a function to generate the N_c codewords in each bin and the associated set of decoding regions $\mathcal{D}$ (with N_c possibly equal to $+\infty$),
- the use of a secret key $\mathbf{k}$ which governs the location of the decoding regions and prevents the access to the watermarking channel.
- If N_m and N_c are large, one has also to find a fast and efficient way to have access to the different bins, codewords and the set $\mathcal{D}$ into a potential high dimensional space $\mathcal{X}$. As we shall see in the next section, the use of lattices is a convenient way to generate bins and decoding regions, and quantisers associated to lattices can be used to perform embedding by efficiently picking the closest codewords (see next subsection). Another convenient way to have access to a high dimensional dictionary is to use trellis (see Sect. 2.4.3).

2.4.1 Distortion Compensated Quantisation Index Modulation (DC-QIM)

Watermarking using Distortion Compensated Quantisation Index Modulation (DC-QIM) was first proposed by Chen and Wornell [9] and has been extended for cubic lattices by Eggers et al. [10] and other classes of lattices by Moulin and Koetter [11].

One bin in Costa's framework is provided by a coarse lattice Λ_c representing a set of points in $\mathcal{X}$. The coarse lattice Λ_c can be defined as

$$\Lambda_c = \{\boldsymbol{\lambda} = G \cdot \mathbf{i} : \mathbf{i} \in \mathbb{Z}^{N_v}\}, \tag{2.4.1}$$

where G is a $N_v \times N_v$ generator matrix and defines a set of point in $\mathcal{X} = \mathbb{R}^{N_v}$. The N_m different bins are generated by using translations of Λ_c. The bin/lattice Λ_m coding the message m is given by ($\Lambda_m = \Lambda_c + \mathbf{d}_m + \mathbf{k}$) where $\mathbf{d}_m$ is called a coset leader and $\mathbf{k}$ is a secret key that allows us to obtain a secret lattice configuration (see Fig. 2.8 for an illustration of these entities) . The coset leaders $\{\mathbf{d}_m\}$ are constructed using a specific lattice construction [12–14]. The most popular are the self-similar constructions, which consists in applying a scaling factor (in this case $\Lambda_f = \alpha\Lambda_c$) and/or a rotation of the coarse lattice Λ_c. "Construction A" consists in generating the set $\{\mathbf{d}_m\}$ inside a unitary hypercube and then shifting the elementary construction over $\mathcal{X}$.

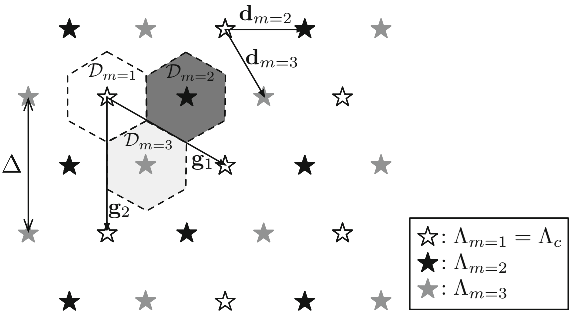


Fig. 2.8 Lattices/Bins for the different messages for $N_m = 3$ and $N_v = 2$ using the coset leaders $\mathbf{d}_{m=2}$ and $\mathbf{d}_{m=3}$ ($\mathbf{d}_{m=1} = \mathbf{0}$). In this example we set $\mathbf{k} = \mathbf{0}$. $\mathbf{g}_1$ and $\mathbf{g}_2$ are the basis vectors of the generator matrix used to generate Λ_c, and $\mathcal{D}_{m=1}$, $\mathcal{D}_{m=2}$, $\mathcal{D}_{m=3}$ represent three decoding regions

The set $\mathcal{D}$ of decoding regions associated to the watermarking scheme are the voronoï cells of the fine lattice Λ_f representing the union of the bins coding all the messages: $\Lambda_f = \cup_{m=1}^{N_m} \Lambda_m$:

$$\mathcal{D} = V(\Lambda_f), \tag{2.4.2}$$

where the function $V(.)$ returns the Voronoï cell of each point of the lattice Λ_f.

One practical way to compute efficiently the nearest decoding region of one point $\mathbf{a}$ associated to the message m is to use a quantiser associated to the bin $\Lambda_m = \Lambda_c + \mathbf{d}_m + \mathbf{k}$ which is equivalent to quantise $\mathbf{a} - \mathbf{d}_m - \mathbf{k}$ according to Λ_c. The first implementation of Quantisation Index Modulation (QIM) simply proposes to embed a message m by quantising the host $\mathbf{x}$ using the appropriate bin Λ_m [9]:

$$e_{QIM}(\mathbf{x}, m, \mathbf{k}) = Q_{\Lambda_c}(\mathbf{x} - \mathbf{d}_m - \mathbf{k}) + \mathbf{d}_m + \mathbf{k}, \tag{2.4.3}$$

the distortion being dependant of the quantisation step Δ, i.e. the distance between two closest elements of Λ_m. Note that the implementation of the quantisation can be straightforward or not, it depends of the lattice Λ_c.

The authors also propose to perform embedding by moving toward the closest point of Λ_m, i.e. moving inside the closest decoding region, using a distortion compensation parameter α. The method is called Distortion Compensated Quantisation Index Modulation (DC-QIM) and the embedding formula is given by:

$$e_{DC-QIM}(\mathbf{x}, m, \mathbf{k}) = \mathbf{x} + \alpha \left(Q_{\Lambda_c}(\mathbf{x} - \mathbf{d}_m - \mathbf{k}) - (\mathbf{x} - \mathbf{d}_m - \mathbf{k}) \right), \tag{2.4.4}$$

which is identical to $e_{QIM}(.)$ for $\alpha = 1$. Note that depending on the lattice, there exists a minimum value α_{min} for which all the host contents are moved inside the appropriate decoding region. For embeddings using values below α_{min}, all the watermarked contents do not embed the correct message since the embedding distortion is not important enough (for $\alpha = 0$ the embedding in completely ineffective since $\mathbf{y} = \mathbf{x}$).

For both QIM and DC-QIM, the decoding function is the same, it consists in finding the bin $\Lambda_{m'}$ which induces the decoding region containing the corrupted vector $\mathbf{z}$ (see Fig. 2.8). Practically this is performed by computing the quantised value of **z** for each bin and by selecting the bin which provides the smallest quantisation error:

$$\hat{m} = d(\mathbf{z}, \mathbf{k}) = \arg\min_{m'} ||Q_{\Lambda_c}(\mathbf{z} - \mathbf{d}_{m'} - \mathbf{k}) - (\mathbf{z} - \mathbf{d}_{m'} - \mathbf{k})||. \tag{2.4.5}$$

2.4.2 Scalar Costa Scheme (SCS)

Eggers et al. [10] have analysed the robustness performance of DC-QIM using a cubic lattice, i.e. using a uniform quantizer applied on each component. The quantisation step being Δ, we have for each component x of $\mathbf{x}$:

$$Q_{\Lambda_c}(x) = \text{sign}\,(x) \cdot \Delta \cdot \lfloor \frac{|x|}{\Delta} + \frac{1}{2} \rfloor, \tag{2.4.6}$$

with $\lfloor . \rfloor$ the floor function. Each coset leader is equal to a translation of $d_m = \Delta \frac{m}{N_m}$ and the embedding becomes:

$$e(x, m, k) = x + \alpha \left(Q_{\Lambda_c} \left(x - \Delta \frac{m}{N_m} - k \right) - \left(x - \Delta \frac{m}{N_m} - k \right) \right), \tag{2.4.7}$$

the scalar k being the secret key picked in the interval $[0, \Delta)$. In this setting the distribution of the host signal x is considered as piecewise uniform, additionally the embedding distortion is very small regarding the host signal, e.g. $\sigma_w^2 \ll \sigma_x^2$.

The distortion of the embedding is given by:

$$\sigma_w^2 = \frac{\alpha^2 \Delta^2}{12}. \tag{2.4.8}$$

Figure 2.9 represents the distribution of watermarked contents for QIM and SCS with a given α_r in 2D ($N_v = 2$) for binary embedding on each component ($N_m / N_v = 2$) for the same embedding distortion DWR $= 20$ dB. We can notice the concentration of the watermarked contents on the quantisation cells for QIM and inside a square of length $(1 - \alpha)\Delta$ for SCS. Because of the structure of the lattice, the decoding regions are cubic.

In order to maximise the robustness, the author compute the achievable rate R of the watermarking channel which is given by the mutual information between the attacked signal and the embedded symbol:

$$R = I(Z, M) = -\int_{\Delta} p_Z(z) \log_2 p_Z(z) dz + \frac{1}{N_m} \sum_{m=1}^{N_m} \int_{\Delta} p_Z(z|m) \log_2 p_Z(z|d) dz, \tag{2.4.9}$$

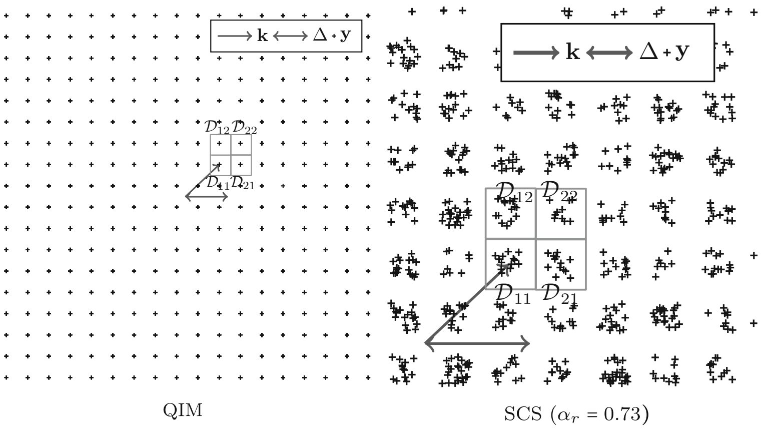


Fig. 2.9 Geometrical view of SCS watermarking for $N_v = 2$ and $N_m/N_v = 2$ (1 bit/sample) for QIM and SCS with 4 decoding regions . DWR = 20 dB

and the authors have derived an approximation of the embedding parameter maximising the achievable rate R for a given WNR. The approximation is given by [10]:

$$\alpha_r = \sqrt{\frac{1}{1 + 2.71 \cdot 10^{-\mathrm{WNR}/10}}}. \tag{2.4.10}$$

For this scheme, the minimum value of α which allows us to watermark contents without decoding error during the embedding is $\alpha_{min} = (N_m - 1)/N_m$. Figure 2.10 shows the robustness gap between QIM and SCS for an AWGN channel according to the WNR for a same embedding distortion (DWR = 20 dB), the Bit Error Rate p_{be} has been practically computed using 10^4 watermarked contents.

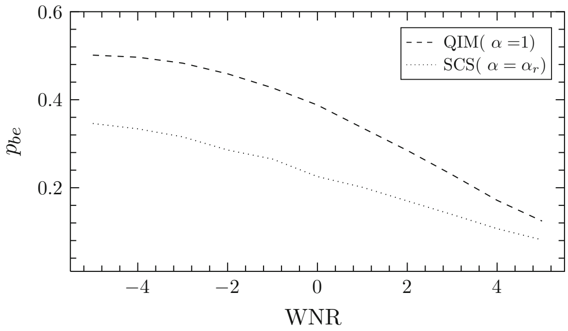


Fig. 2.10 Practical comparison between the BER for QIM and SCS w.r.t. the WNR ($N_v = 1, N_m = 2$, DWR = 20 dB) for an AWGN channel. The empirical probabilities are computed using 10^4 contents. ⇝ **implementation: run `BERSCSPrac.R`**

2.4.3 Trellis-Based Watermarking

The use of trellis for watermarking is a practical way to perform dirty paper coding [7]. Dirty paper coding implies the use of a codebook $\mathcal{C}$ of codewords with a mapping between codewords and messages. Dirty Paper Trellis (DPT) codes have two main assets: the generation of the codebook $\mathcal{C}$ is systematic and the search for the nearest codeword can be efficiently performed with a Viterbi decoder [15]. A DPT is a network defined by several parameters:

1. the number of states N_s,
2. the number of arcs per state N_a,
3. the N_v-dimensional pseudo-random patterns associated to each one of the $N_a \cdot N_s$ arcs, which can be assimilated to the carrier used in spread spectrum schemes,
4. the binary label associated to each one of the $N_a \cdot N_s$ arcs,
5. the number of steps in the trellis, corresponding in [8] to the number of bits N_b of the trellis,
6. the connectivity between the states, i.e. for each arc the labels of the two states which are connected.

Figure 2.11 depicts an example of a DPT. One can notice that the configuration of the trellis is simply repeated from one step to another. Moreover, the number of outgoing and incoming arcs per state is constant. These are common assumptions in trellis coding.

A DPT is thus associated with a codebook $\mathcal{C} = \{\mathbf{c_i}, i \in [1, \ldots, N_s \cdot N_a^{N_b}]\}$ of $N_s \cdot N_a^{N_b}$ codewords in a $N_v \cdot N_b$-dimensional space. Each codeword can be built in such a way that it corresponds to a path in the trellis and encodes N_b bits. This message can be retrieved by concatenating the binary labels of the arcs along the corresponding path.

DPT watermarking makes use of both *informed coding* and *informed embedding* [8]. Informed coding consists in selecting the codeword $\mathbf{g}$ in the codebook $\mathcal{C}$ that is the closest to the host vector $\mathbf{x}$ and that encodes the desired message. This is

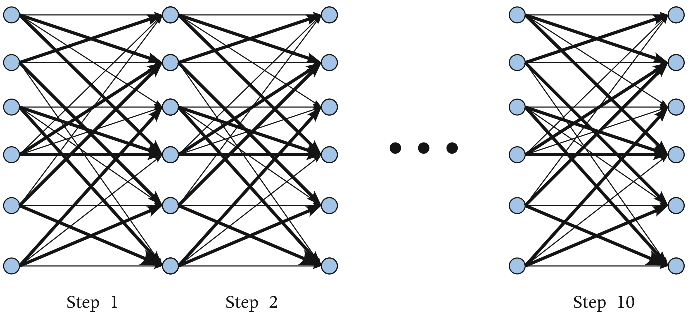


Fig. 2.11 Example of the structure of a 10 steps trellis with 6 states and 4 arcs per states. *Bold* and normal arcs denote respectively 0 and 1 valued labels

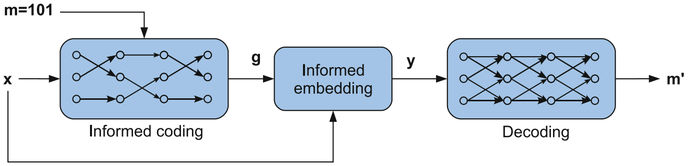


Fig. 2.12 Principles of informed embedding and informed coding using trellis codes

the same principle than for dirty paper coding. The selection is done here by running a Viterbi decoder with an expurgated trellis containing only arcs whose binary labels are in accordance with the message to be embedded. As a result, any path through the trellis encodes the desired message. The Viterbi decoder is then used to maximize/minimize a given function i.e. to find the *best* codeword in this subset according to some criterion. In their original article [8], the authors proposed to keep the codeword with the highest linear correlation with the host vector **x**.

At this point, informed embedding is used to reduce the distance between the host vector **x** and the selected codeword **g**. It basically computes a watermarked vector **y** that is as close as possible from **x** while being at the same time within the detection region of the desired codeword **g** with a guaranteed level of robustness to additive white Gaussian noise (AWGN). In practice, a sub-optimal iterative algorithm is used combined with a Monte-Carlo procedure to find this watermarked vector **y** [8].

On the receiver side, the embedded message is extracted by running a Vitterbi decoder with the whole DPT. The optimal path is thus identified and the corresponding message retrieved by concatenating the binary label of the arcs along this path. The whole procedure is illustrated in Fig. 2.12.

2.5 Conclusions of the Chapter

We can see from this chapter that, once the watermarking scheme is fully described, it is possible to compute the distortion and the robustness of the scheme, either using analytical formulas or by practically measuring the visual impact and the symbol error rate after a given process. In the next chapter we show how to measure the third constraint of the watermarking game, namely security, and how to distinguish different scheme w.r.t. their security classes.

References

1. Kalke T (2001) Considerations on watermarking security. In: Proceedings of the of MMSP. Cannes, France, October pp 201–206
2. Hartung F, Su JK, Girod B (1999) Spread spectrum watermarking: malicious attacks and counter-attacks. In: San Jose CA, January S (eds) Proceedings of SPIE: Security and watermarking of multimedia contents Bd. 3657. pp 147–158

3. Florencio M (2003) Improved spread spectrum: a new modulation technique for robust watermarking. In: IEEE Transactions on Signal Processing 51
4. Furon T, Jégourel C, Guyader A, Cérou F (2009) Estimating the probability fo false alarm for a zero-bit watermarking technique. In: Digital Signal Processing, 2009 16th International Conference on IEEE, pp 1–8
5. Miller ML, Cox IJ, Bloom JA (2000) Informed embedding: exploiting image and detector information during watermark insertion. In: Image Processing, 2000. Proceedings. 2000 International Conference on Bd. 3 IEEE, p 1–4
6. Furon T, Bas P (2008) Broken arrows. EURASIP J Inf Secur 1–13:1687–4161
7. Costa M (1983) Writing on dirty paper. In: IEEE Trans. on Inf Theor. 29:439–441
8. Miller ML, Doërr GJ, Cox IJ (2004) Applying informed coding and embedding to design a robust, high capacity watermark. In: IEEE Trans. on Imag Process. 6:791–807
9. Chen B, Wornell GW (2001) Quantization index modulation : a class of provably good methods for digital watermarking and information embedding. IEEE Trans Inf Theor 47(4):1423–1443
10. Eggers JJ, Buml R, Tzschoppe R, Girod B (2003) Scalar Costa Scheme for Information Embedding. In: IEEE Trans Signal Process. 51: 1003–1019
11. Moulin P, Koetter R (2005) Data hiding codes. In: Proceedings of IEEE 93, December 12 pp 2083–2126
12. Conway JH, Sloane NJA (1998) Sphere packings, lattices and groups. Springer, New York
13. Erez U, Litsyn S, Zamir R (2005) Lattices which are good for (almost) everything. IEEE Trans Inf Theor 51:3401–3416
14. Pérez-Freire L, Pérez-González F (2008) Security of lattice-based data hiding against the watermarked only attack. In: IEEE Trans Inf Forensics Secur 3, 4:593–610. http://dx.doi.org/10.1109/TIFS.2008.2002938. doi:10.1109/TIFS.2008.2002938. ISSN 1556–6013
15. Andrew JV (1995) CDMA: Principles of Spread Spectrum Communication. Addison-Wesley, Boston ISBN 0–201–63374–4

http://www.springer.com/978-981-10-0505-3

Watermarking Security
Bas, P.; Furon, T.; Cayre, F.; Doërr, G.; Mathon, B.
2016, IX, 125 p. 70 illus., 29 illus. in color., Softcover
ISBN: 978-981-10-0505-3