Practical Security Analysis of Dirty Paper Trellis Watermarking

Patrick Bas, Gwenaël Doërr

Practical Security Analysis of Dirty Paper Trellis Watermarking

Patrick Bas[1,2] and Gwenaël Doërr[3]

[1] CIS / Helsinki University of Technology
P.O. Box 5400, FI-02015 HUT Finland

[2] Gipsa-lab CNRS INPG
961, rue de la Houille Blanche BP 46
38042 St. Martin d'Hères, France

[3] University College London
UCL Adastral Park – Ross Building 2
Martlesham IP5 3RE , United Kingdom

Abstract. This paper analyses the security of dirty paper trellis (DPT) watermarking schemes which use both informed coding and informed embedding. After recalling the principles of message embedding with DPT watermarking, the secret parameters of the scheme are highlighted. The security weaknesses of DPT watermarking are then presented: in the watermarked contents only attack (WOA) setup, the watermarked data-set exhibits clusters corresponding to the different patterns attached to the arcs of the trellis. The K-means clustering algorithm is used to estimate these patterns and a co-occurrence analysis is performed to retrieve the connectivity of the trellis. Experimental results demonstrate that it is possible to accurately estimate the trellis configuration, which enables to perform attacks much more efficient than simple additive white Gaussian noise (AWGN).

1 Introduction

Beside conventional measurements of performances such as robustness to channel transmission, receiver operating characteristics (ROC) curves or imperceptibility, *security* has recently been acknowledged to be also of fundamental importance in digital watermarking. By definition, security oriented attacks "aim at gaining knowledge about the secrets of the system (e.g. the embedding and/or the detection keys)" [1]. In practice, it implies that if the security of a scheme is compromised, different attacks such as message modification, message copy or message erasure are possible while keeping a very low distortion. Hence, watermarking schemes need to be carefully analysed to identify its *security level*, e.g. the number of contents that are needed to estimate accurately the secret key [2].

Security of watermarking schemes can be assessed either with a theoretical analysis or with a practical evaluation. Theoretical security analysis consists

in calculating the information leakage occurring when observing several watermarked contents by means of information theoretic measures such as equivocation or mutual information between the secret key and the observations [2, 1, 3]. These measurements prove whether or not there is some information leakage that might be exploited to estimate the secret key. However, they do not give any clue about the tools that could be used to perform this estimation.

On the other hand, practical security analysis consists in designing attacks which make possible to estimate the secret parameters used during embedding. Only a few attempts in this direction have been reported so far and they have mostly focused on basic watermarking schemes. For example, in [4–6], the authors propose different blind source separation methods to estimate secret patterns that are used in spread-spectrum or spread transform dither modulation schemes for both independent and identically distributed (iid) and non-iid signals. In [3], the authors adopt a set-membership approach to estimate the dither vector used during DC-DM embedding.

This paper proposes a practical security analysis of dirty paper trellis (DPT) watermarking schemes, which have been proven to achieve high performances with respect to robustness and payload [7]. Section 2 first recalls the principles of DPT watermarking. In Section 3, the different parameters that define the secret key are identified, and a worst case attack (WCA) relying on the estimation of the secret key is introduced. In Section 4, practical tools are proposed to estimate each parameter of the trellis, namely the patterns attached to the arcs and the configuration of the trellis. Section 5 reports the performances of the WCA according to both the embedding distortion and the number of observed contents. Finally, some perspectives to improve the security of DPT watermarking schemes are presented in Section 6.

2 Dirty Paper Trellis Watermarking

2.1 Notations and Parameters Definition

In this paper, the host vector is denoted $\mathbf{x}$ and the watermarked vector $\mathbf{y}$. The latter one carries a N_b bits message $\mathbf{m}$. Each bit of the message is encoded on N_v coefficients and therefore $\mathbf{x}$ and $\mathbf{y}$ are both $N_b \cdot N_v$-dimensional vectors [4]. Moreover, $||\mathbf{v}||$ denotes the Euclidian norm of the vector $\mathbf{v}$ and $\mathbf{v}(k)$ the k^{th} component of $\mathbf{v}$. Finally, embedding distortions are given using the watermark to content ratio (WCR) expressed in decibels.

2.2 Trellis-based watermarking

The use of trellis for watermarking is a practical way to perform dirty paper coding [8]. Dirty paper coding implies the use of a codebook $\mathcal{C}$ of codewords

[4] Note that an attacker will have the opportunity to observe N_o watermarked contents. Practical values of N_o can go from 1 (a single image for example) to several thousands (a set of videos where each frame carries a payload).

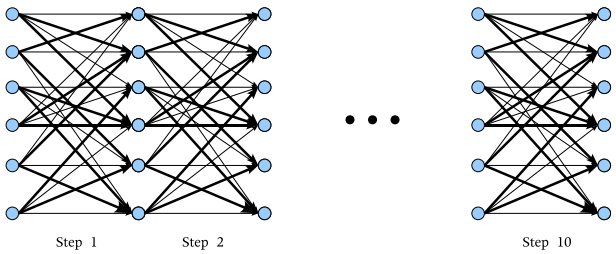


Fig. 1. Example of the structure of a 10 steps trellis with 6 states and 4 arcs per states. Bold and normal arcs denote respectively 0 and 1 valued labels.

with a mapping between codewords and messages. The key difference with conventional codes is that different codewords can map to the same message. This defines a coset $\mathcal{C}_m$ of codewords for each message $\mathbf{m}$. The watermarking process then reduces to (i) identify the codeword in the coset $\mathcal{C}_m$, related to the message to be hidden, which is the nearest from the host vector $\mathbf{x}$, and (ii) move the host vector inside the detection region of the selected codeword. According to Costa's framework, using this setup the capacity of the channel does not depend on the host $\mathbf{x}$.

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 [9]. A DPT is defined by several parameters:

1. the number of states N_s,
2. the number of arcs per state N_a,
3. the connectivity between the states i.e. in which state an arc starts and in which state it ends,
4. the N_v-dimensional pseudo-random patterns associated to each one of the $N_a.N_s$ arcs, which can be assimilated to the carrier used in spread spectrum schemes,
5. the binary label associated to each one of the $N_a.N_s$ arcs,
6. the number of steps N_b in the trellis.

Figure 1 depicts an example of a DPT. One can notice that the configuration of the trellis is simply repeated from one step to another without any change. 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 $\mathcal{C} = \{\mathbf{c_i}, i \in [1, ..., N_s \cdot N_a^{N_b}]\}$ of $N_s \cdot N_a^{N_b}$ codewords in a $N_v \cdot N_b$-dimensional space. Each codeword corresponds to a path in the trellis and encodes a N_b bits message. This message can be retrieved by concatenating the binary labels of the arcs along the corresponding path.

DPT watermarking makes makes use of both *informed coding* and *informed embedding* [7]. 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

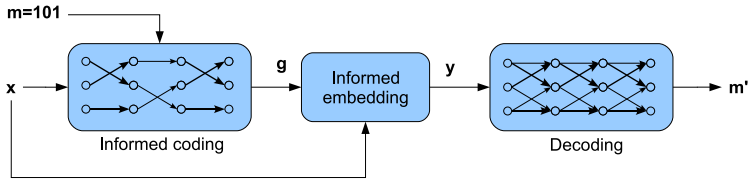


Fig. 2. Main principles of DPT watermarking. In this example $N_b = 3$, $N_s = 3$, $N_a = 2$. Three alternative codewords are available in the expurgated trellis to represent the message $\mathbf{m}$. The codeword $\mathbf{g}$ with the highest correlation with $\mathbf{x}$ is identified using the Viterbi algorithm. Afterward, the watermarked vector $\mathbf{y}$ is computed taking into account $\mathbf{g}$. On the receiver side, the detector uses the whole DPT to retrieve the embedded payload.

message. The selection is done 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 [7], the authors proposed to keep the codeword with the highest linear correlation with the host vector $\mathbf{x}$.

At this point, informed embedding is used to reduce the distance between the host vector $\mathbf{x}$ and the selected codeword $\mathbf{g}$. It basically computes a watermarked vector $\mathbf{y}$ that is as close as possible from $\mathbf{x}$ while being at the same time within the detection region of the desired codeword $\mathbf{g}$ 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 $\mathbf{y}$ [7].

On the receiver side, the embedded message is extracted by running a Viterbi 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 Figure 2.

3 DPT Secret Key and Worst Case Attack

First, some parameters of the DPT will be assumed to be public. It may not always be true in practice, but usually these parameters are fixed according to the desired robustness or payload of the algorithm. In this study for instance, the three parameters N_s, N_a and N_b will be known.

Furthermore, processed contents will be assumed not to be shuffled before embedding i.e. they are directly watermarked without prior permutation of the samples position. The problem of inverting a hypothetical shuffle relies on the security of the shuffle itself and is far beyond the scope of this paper.

To define the secret key relative to a DPT watermarking scheme, it is necessary to identify which information is required by the attacker to perform security-oriented attacks such as:

- decoding the embedded message,
- altering the embedded message while producing the minimal possible distortion,
- copying the message to another content while producing the minimal possible distortion.

To decode the embedded message, the previous section recalls that all parameters of the DPT are needed. This includes by definition the patterns attached to the arcs, the connectivity between the states, and the binary labels of the arcs. To copy a message in an optimal way, it is first necessary to decode them and then to embed them into another content. Therefore, the same parameters are required. On the other hand, to alter the embedded message, all previous parameters are needed except the binary labels. Indeed, the watermarked vector $\mathbf{y}$ only need to be moved toward another vector $\mathbf{y}_{\mathrm{A}}$ so that it no longer lies in the decoding region of $\mathbf{g}$. As long as a *neighbour* codeword is selected, it is unlikely to encode the same message and it will be close enough to avoid large distortion. This threat can be seen as the worst case attack (WCA) for DPT watermarking [10].

To perform this attack, it is necessary to know the two closest codewords from the watermarked vector $\mathbf{y}$, i.e. the embedded codeword $\mathbf{g}$, and the second closest codeword from $\mathbf{y}$ ($\mathbf{b}_1$ in Figure 3 (a)). The attacker simply needs then to move the watermark content $\mathbf{y}$ somewhere inside the decoding region of this second best codeword ($\mathbf{y}_{\mathrm{A}}$ in Figure 3 (a)) to make the detector fail while minimizing the distortion of the attack. In practice, this second best codeword is identified by feeding the Viterbi decoder with the watermarked vector $\mathbf{y}$ and successively forbidding a single step of the optimal path $\mathbf{g}$. This results in N_b candidates and the closest to $\mathbf{y}$ is retained as the second best codeword to be used in the WCA. This procedure is depicted in Figure 3 (b).

4 DPT Parameters Estimation

For the sake of generality, the analysis will be done according to the Watermarked content Only Attack (WOA) setup [2], where the attacker observes different contents watermarked with different messages using the same secret key. The aim of this section is to present different techniques that can be used to estimate different parameters of a DPT that constitute the secret key, namely the patterns, the connectivity and the binary labels of the arcs.

4.1 Side Effects of Informed Embedding

Let $\mathcal{U} = \{\mathbf{u}_i, i \in [1, ..., N_a \cdot N_s]\}$ be the set of patterns, also referred to as carriers, associated with the arcs of the DPT. In practice, each pattern is usually

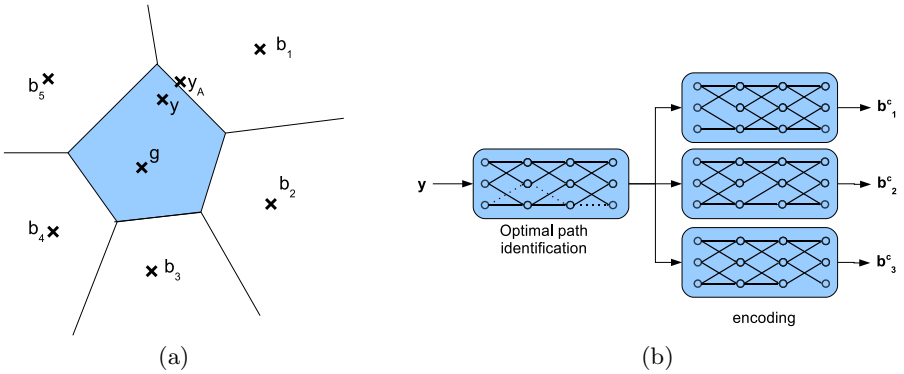


Fig. 3. Worst case attack for DPT watermarking. (a): To be optimal, the attacker needs to find the closest point to $\mathbf{y}$ outside the detection region of the embedded codeword $\mathbf{g}$ (grey area). To do so, he needs to identify the second nearest codeword from $\mathbf{y}$, i.e. $\mathbf{b}_1$ in the Figure. (b): To identify the second best codeword, the Viterbi decoder is run several times with a single step of the optimal trellis forbidden. The codeword amongst the N_b candidates which is the closest to $\mathbf{y}$ is retained.

normalised, e.g. $||\mathbf{u}_i|| = 1, \forall i$. As a result, each pattern can be seen as a point on the surface of the N_v-dimensional unit sphere. Moreover, each codeword $\mathbf{c}_i$ of the DPT is a $N_v \cdot N_b$-dimensional vector of norm $\sqrt{N_b}$ and can be considered as a point on the surface of a $N_v \cdot N_b$-dimensional sphere of radius $\sqrt{N_b}$, denoted $\mathcal{S}_c$.

Viterbi decoding aims at finding the codeword $\mathbf{c} \in \mathcal{C}$ which is the most correlated with some input vector $\mathbf{v}$, i.e. it evaluates and maximises:

$$\text{corr}(\mathbf{c}_i, \mathbf{v}) = \frac{< \mathbf{c}_i, \mathbf{v} >}{N_b \cdot N_v} = \frac{\sum_{j=1}^{N_b \cdot N_v} \mathbf{c}_i(j)\mathbf{v}(j)}{N_b \cdot N_v}, \tag{1}$$

which is equivalent to:

$$\text{corr}(\mathbf{c}_i, \mathbf{v}) = \frac{||\mathbf{v}||.||\mathbf{c}_i|| \cos(\theta_i)}{N_b \cdot N_v}, \tag{2}$$

where θ_i denotes the angle between $\mathbf{v}$ and $\mathbf{c}_i$. Because $||\mathbf{v}||$, $||\mathbf{c}_i||$ and $N_b \cdot N_v$ are constant terms, the codeword that is selected basically maximises $\cos(\theta_i)$.

In other words, the Viterbi decoder returns the codeword which is at the smallest angular distance from the input vector. This implies that when one wants to embed a codeword $\mathbf{g}$ in a host vector $\mathbf{x}$, it is necessary to produce a watermarked vector $\mathbf{y}$ whose angular distance with $\mathbf{g}$ is lower than with any other codeword in $\mathcal{C}$. Moreover, the higher the robustness constraint, the closer the watermarked contents to the desired codeword. Consequently, considering the distribution of normalized observations $\mathbf{y}^* = \mathbf{y}/||\mathbf{y}||$ one might observe *clusters* corresponding to the codewords in $\mathcal{C}$ on the surface of the $N_v \cdot N_b$ dimensional sphere $\mathcal{S}_c$.

4.2 Patterns Estimation using a Clustering Method

Data clustering algorithms enable to analyse a large set of data by partitioning the set into subsets called clusters. Clusters are build such as to minimize the average distance between each data point and the nearest cluster center, also referred to as centroid. Given k the number of clusters, a clustering algorithm also returns the label of the centroid associated with each data point.

K-means algorithm. In this work, the K-means algorithm has been used to provide a partition of the observed space. This algorithm labels each data point to the cluster whose centroid is the nearest. The centroid is defined as the center of mass of the points in the cluster, and its coordinates are given by the arithmetic mean of the coordinates of all the points in the cluster.

The implemented version of the algorithm was proposed by MacQueen [11] and is described below:

1. Choose k the number of clusters,
2. Initialise the centroids,
3. Assign each data point to the nearest centroid,
4. Update the centroid coordinates,
5. Go back to step 3 until some convergence criterion is met.

This algorithm is easy to implement, fast, and it is possible to run it on large datasets. However K-means does not yield the same result for each run, i.e. the final clusters depend on the initial random assignments. One solution to overcome this problem is to perform multiple runs with different initialisations and to keep the result which provides the lowest intra-cluster variance. To ensure that the initial clusters are evenly distributed over the data set, a random initialisation using the KZZ method [12] has been used.

Definition of the dataset. A segment $\mathbf{s}$ is a portion of the observed watermarked vector $\mathbf{y}$ corresponding to a single step. Therefore, $\mathbf{s}$ is of size N_v and $\mathbf{y}$ is composed of N_b segments. Two alternative strategies are possible to estimate the secret parameters of the DPT:

1. Apply the K-means algorithm to estimate the centroids representing the codewords of the trellis. Then it has to find $k = N_s \cdot N_a^{N_b}$ centroids in a $N_v \cdot N_b$-dimensional space using a dataset of normalised watermarked vectors.
2. Apply the K-means algorithm to estimate the centroids representing the patterns of the trellis. Then it has to find $k = N_s \cdot N_a$ centroids in a N_v-dimensional space using a data-set of normalised watermarked segments.

Observing N_o watermarked contents is equivalent to observing $N_o \cdot N_b$ watermarked segments. As a result, the two strategies proposed earlier involve respectively $\frac{N_o}{N_s \cdot N_a^{N_b}}$ and $\frac{N_o \cdot N_b}{N_s \cdot N_a}$ observations per centroid. In other words, the second solution provides $N_b \cdot N_a^{N_b - 1}$ times more observations per centroid than

the first one to perform clustering. This problem is related to the *curse of dimensionality*, well known in machine learning, which states that the number of observations needed to learn topological objects such as clusters is exponential with respect to the dimension of the problem. Since the main concern here is the estimation of the patterns used in the DPT, the second solution is preferred to improve the estimation accuracy for the same number of observed contents.

Analysis of estimation accuracy according to distortion. The accuracy of the estimated patterns is inherently related to the embedding distortion, and therefore with the robustness constraint. Figure 4 depicts two typical examples of 3D distributions of normalised watermarked segments for two different embedding distortions. In this case only 6 codewords are used and one bit is embedded. The yellow balls indicate the centroids estimated using the K-means algorithm and the grey balls the position of the true patterns. In this example, patterns are chosen to be either orthogonal or collinear (the set of orthogonal patterns is multiplied by -1 to obtain collinear ones). Each point of the distribution has a color depending on the center it has been associated.

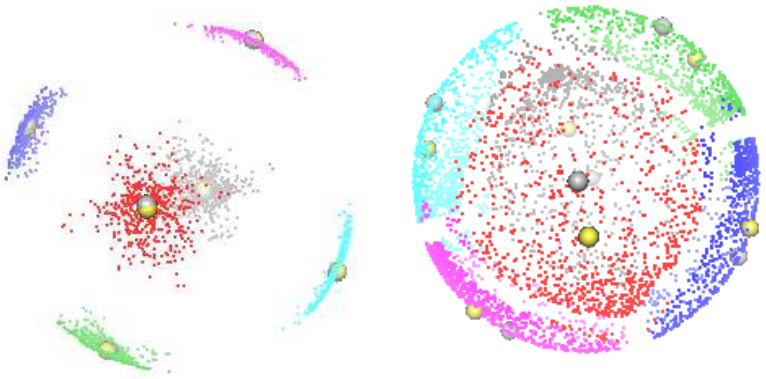
In each case, the detection regions are represented by clusters which are clearly identifiable. Moreover the mapping between detection cells and embedded contents is consistent. However, for the smallest distortion (WCR=-6.6 dB), watermarked vectors are not uniformly distributed inside the embedding region. This is due to the fact that if two neighbour codewords encode the same message, their border region will have a density of codewords less important than if they encode different messages. This uneven distribution of watermarked codewords in each detection region results in a erroneous estimation of the codeword, the cluster center being "attracted" by the dense borders as illustrated on the right-hand distribution.

Figure 5 shows the accuracy of the DPT patterns estimation in the case of a realistic watermarking scenario. The different parameters of the trellis are defined here by $N_v = 12$, $N_b = 10$, $N_s = 6$, $N_a = 4$, which means that the clustering algorithm has to estimate 24 patterns of 12 samples each[5]. To evaluate the estimation process, the average of the difference between the two largest normalised correlations between each real and estimated patterns for each pattern is computed i.e.:

$$\Delta = \frac{1}{N_s \cdot N_a} \sum_i \left[\max 1_j(\mathrm{corr}_N(\mathbf{cl}_i, \mathbf{u}_j)) - \max 2_j(\mathrm{corr}_N(\mathbf{cl}_i, \mathbf{u}_j)) \right] \qquad (3)$$

where $\mathbf{cl}_i$ is the estimated centroid of the i^{th} cluster, corr_N denotes the normalised correlation, and $\max 1_j$ (resp. $\max 2_j$) represents the first (resp. second) value when an array is sorted by descending order. As previously, the set of 24 patterns are orthogonal or collinear between themselves. As a result, Δ is equal to one if the estimation of each pattern is perfect and decreases with respect to the accuracy of estimations.

[5] $N_v = 12$ is the number of DCT coefficients that are used in the image watermarking scheme presented in [7].



WCR= 1.5dB, $N_b = 1$ WCR= -6.6dB, $N_b = 1$

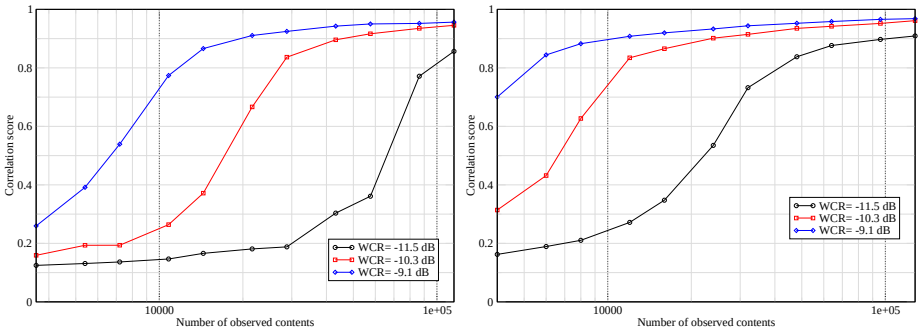
Fig. 4. Distributions of normalised watermarked contents ($N_v = 3$, $N_b = 1$, $N_s = 3$, $N_a = 2$, $N_o = 5000$). Locations of the real (gray) and estimated (yellow) patterns using the K-means algorithm. The visualization is easier using a color output.

The difference between the two highest correlation score magnifies the contrast between accurate and non-accurate estimation of the DPT patterns when they are orthogonal or opposite. A score Δ close to 1 means that each estimated pattern is much closer to one of the original patterns than the others. On the other hand, a score Δ close to 0 means that each estimated pattern is equally distant from two original patterns. Consequently, the estimation of the original patterns is not possible. Using only max1() would have decreased the difference between accurate and non-accurate estimations because even random patterns may have an important correlation with fixed ones if N_v is low.

The evolution of the estimation accuracy with respect to different embedding distortions and different number of observations is given in Figure 5 for observations composed of either one or two segments. If the considered dataset is composed of couples of segments, the number of observations necessary to obtain the same accuracy than for one segment is roughly multiplied by 4. This confirms the "curse of dimensionality" effect mentioned earlier. Moreover, as expected, the estimation accuracy increases with the number of observed contents and the embedding distortion i.e. the robustness constraint. While more than 128000 observations are needed to obtain an accuracy of 0.9 with $WCR = -11.5dB$ and a data set of single segments, 24000 and 8000 observations are needed respectively for $WCR = -10.3dB$ and $WCR = -9.1dB$.

4.3 Note on Label Estimation

As mentioned in Section 3, the estimation of the binary label associated to each arc is not possible in the WOA framework. Note however that, for the Known Message Attack scenario (KMA) where each embedded message is known [2],



Observation: couple of segments
96 clusters in a 24-D space

Observation: single segment
24 clusters in a 12-D space

Fig. 5. Accuracy of the DPT patterns estimation ($N_v = 12$, $N_b = 10$, $N_s = 6$, $N_a = 4$). Average after 10 trials. For each trial, 10 K-means runs are performed.

the binary labels can easily be estimated by examining the bits associated to each segment. For an estimated centroid, the binary label will be determined as the most frequent bit related to the segments within the cluster.

Another way to deal with this issue is to use supervised clustering techniques such as Learning Vector Quantization [13]. This approach might be more efficient than K-Means since it considers the class of observations as a-priori information.

4.4 Connections and State Estimation

In the DPT estimation process, the next step is to estimate the connectivity of the trellis. This can be seen as learning which patterns are emitted at step $t+1$ knowing that a given pattern has been emitted at step t. This estimation can be done by using a co-occurrence matrix $\mathbf{C}$ which is a square matrix of size $N_s \cdot N_a$. Each element $\mathbf{C}(i,j)$ of the matrix is expressed by:

$$\mathbf{C}(i,j) = \mathrm{occ}(\mathbf{s}_t \in \mathcal{C}\updownarrow_i, \mathbf{s}_{t+1} \in \mathcal{C}\updownarrow_j) \tag{4}$$

where $\mathcal{C}\updownarrow_k$ denotes the set representing the k^{th} cluster and $\mathrm{occ}(A,B)$ is an occurrence function that counts the number of times both A and B are true. The test $(\mathbf{s}_t \in \mathcal{C}\updownarrow_i)$ is performed using the classification results of the K-means algorithm used for the patterns estimation. As a result, if the pattern i has been emitted at step t, the N_a maximum values in the i^{th} row of the co-occurrence matrix $\mathbf{C}$ indicate the index of the patterns that can be emitted at step $t+1$. This method implicitly assumes that a different pattern is attached to each arc in the trellis. Therefore, it will fail to deal with the recent improvements proposed for DPT watermarking based on trellis coded modulation [14].

Using the established co-occurrence matrix, it is possible to check whether the estimated connectivity matches the one of the original trellis. For each line i in the matrix, the index of the N_a highest elements are retrieved. As stated

before, each index points to the pattern that can be emitted at step $t+1$ when the pattern i has been emitted at step t. This leads to $N_s \cdot N_a^2$ possible couple of patterns, that can be referred to as *connections*. The connection error rate is then defined as the ratio of connections which are actually not allowed by the original trellis. The lower the connection error rate, the more accurate the estimated connectivity. As depicted in Figure 6, the accuracy relies again on the embedding distortion and the number of observed contents. It should be noted that the number of observed contents necessary to achieve a good estimation of the connections is of the same order of magnitude than for the estimation of patterns.

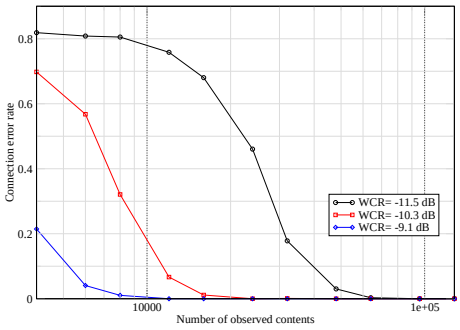


Fig. 6. Connection error rate ($N_v = 12$, $N_b = 10$, $N_s = 6$, $N_a = 4$). Observation: single segment. Average after 10 trials. For each trial, 10 K-means runs are performed.

At this point, using the co-occurrence matrix, it is possible to identify for each pattern, which can also be viewed as an arc, which are the incoming and outgoing states. Each state is estimated up to a permutation with respect to the original trellis. However, this permutation does not hamper the ability of the decoder to retrieve the correct succession of patterns.

All the arcs going toward a given state will give similar rows in the co-occurrence matrix $\mathbf{C}$. Indeed, the rows indicate the choice of patterns that can be emitted afterward when an arc is traversed. Same rows implies same choice i.e. for all these arcs, the same state has been reached. To deal with the potential noise in the co-occurence matrix, a K-means algorithm is run on the rows of $\mathbf{C}$ to identify N_s clusters. Each row is then labeled in accordance to the cluster it belongs to. This label indicates the outgoing state when an arc is traversed i.e. when a given pattern is emitted. For instance, in Figure 7, if the third pattern is emitted, the systems reaches state 1 and can only emit the patterns 1 and 4.

One can then build an outgoing state matrix: it is a simple matrix with entries at the estimated connection index which indicates the outgoing state when the pattern i is emitted at step t. An example is given in Figure 7. This matrix can be read row by row: if the pattern 3 is emitted at step t, then the system is in the third row and one can see that the state 1 is reached. Moreover, this

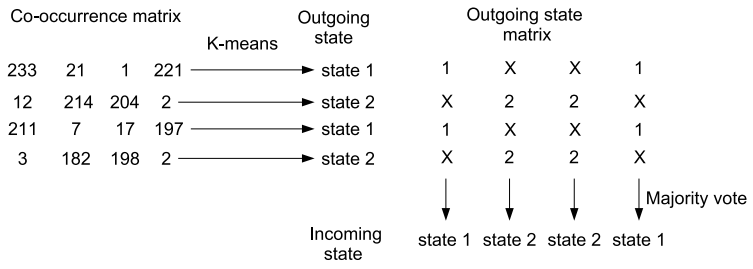


Fig. 7. Example of incoming and outgoing state estimations.

outgoing state matrix can also be read column by column: if the pattern 3 has been emitted at step $t+1$, then the system is in the third column and the entries indicates the possible states of the system before the emission of the pattern i.e. the incoming state. A simple majority vote along each column to accommodate for potentially noisy observations gives then the most likely incoming state for each pattern. In the example depicted in Figure 7, one can see for instance that the pattern 3 is coming from state 2 and is going toward state 1.

5 Results on the worst case attack

Once all the secret parameters of the DPT have been estimated, it is possible to perform the WCA described in Section 3. Results are plotted on Figure 8 for the same setup than previously: $N_v = 12$, $N_b = 10$, $N_s = 6$, $N_a = 4$ and three different embedding distortions. Two different scores are computed to assess the efficiency of the WCA: the classical bit-error rate (BER) and the message error rate (MER). The MER is the most meaningful score because it measures the ability of the WCA to move the watermarked vector $\mathbf{y}$ outside the detection region of the embedded codeword. The BER plot in Figure 8 highlights the fact that the WCA does not necessarily yield the same BER for different embedding distortions once the estimation of the trellis is accurate enough. Indeed, for different distortions, the second best codeword may be different and thus induce a different BER.

The security level s can be defined as the number of watermarked contents necessary to perform a successful WCA e.g. with a MER close to 100%. The values of s for different embedding distortions are reported in Table 1. This table also provides a comparison of the average watermarked signal to noise ratios (SNR) for the WCA and AWGN required to yield equivalent BER. The WCA induces a SNR that is between 12 dB and 14 dB more important than for AWGN (the comparison between MERs would have been even more dramatic).

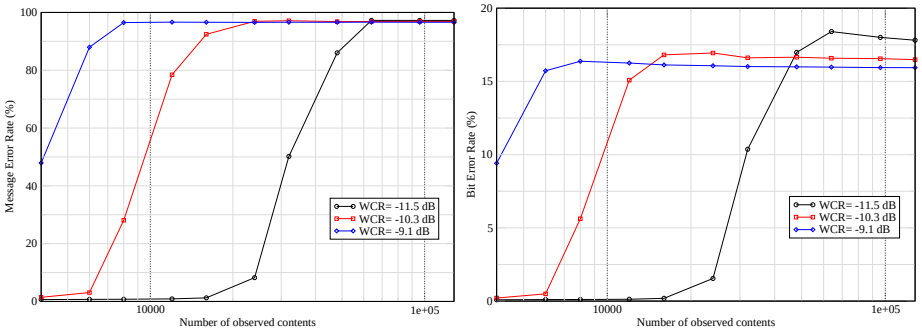


Fig. 8. Message Error Rate and Bit Error Rate after the WCA ($N_v = 12$, $N_b = 10$, $N_s = 6$, $N_a = 4$). Average after 10 trials. For each trial, 10 K-means are performed.

Watermark to Content Ratio	−11.5 dB	−10.3 dB	−9.1 dB
security level s	$64\ 10^3$	$24\ 10^3$	$8\ 10^3$
SNR for the WCA	16.9 dB	16.9 dB	16.9 dB
SNR for AWGN	4.5 dB	3.5 dB	2.4 dB

Table 1. Comparison of the security level and the signal to noise ratio of the WCA and AWGN for equal BER. For SNR, an accurate estimation of the trellis ($N_o = 124000$) is performed.

6 Conclusion and perspectives

This paper has investigated security issues for DPT watermarking schemes. Different properties of this class of schemes have to be highlighted:

- Using the WOA setup, it is impossible to estimate the binary labels associated with each arc of the trellis and consequently it is impossible to copy the message embedded in one content to another one without introducing unacceptable distortion. This property relies on the fact that coding is informed i.e. it is dependent of the host signal. Note that this property is not true for classical Spread Spectrum [5].
- The WOA setup enables however to perform a WCA for this scheme. Machine learning techniques can be used to identify clusters that are created in the data set during the embedding. This estimation has been performed using a K-means algorithm. Different tests suggest that an accurate estimation of the trellis is possible but depends on two parameters: the number of observations and the embedding distortion which is directly linked with the robustness of the scheme.

The assumptions made in this paper on the trellis structure may first look restrictive but encompass a large variety of practical implementations:

- The trellis structure was the same for each step. This hypothesis is important if one want to deal with synchronisation problems. Moreover, if it is not the

case, because the trellis structure is the same for each content in the WOA setup, it is still possible to observe at least N_o similar segments (instead of $N_o \cdot N_b$) and to estimate the patterns for each step.
- The number of outgoing and incoming arcs per state was assumed to be constant. Nevertheless the presented connection and state estimation algorithms can also be used if the arcs change from one step to another.
- The WCR considered are was the order of -10 dB. In the case of smaller WCRs (around -20 dB) either other clustering techniques or a more important number of observations would be necessary. Nevertheless a WCR around -10dB on a dedicated subspace, like medium frequency DCT coefficients for example, is practically realistic.
- In a more the general setup some additional techniques would be required to estimate N_s and N_a for each step, one possibility would be to use hierarchical clustering algorithms to estimate these parameters [15].

Our future works will be focused on the design of secure DPT watermarking schemes. One solution might be to perform the embedding in such a way that the distribution of codewords is similar to the distribution of secure but non-informed coding schemes such as circular watermarking [16].

7 Acknowledgments

Dr. Patrick Bas is supported (in part) by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT and the National French projects ACI-SI Nebbiano, RIAM Estivale and ARA TSAR. Dr Gwenaël Doërr's research is supported in part by the Nuffield Foundation through the grant NAL/32707.

References

1. Comesaña, P., Pérez-Freire, L., Pérez-González, F.: Fundamentals of data hiding security and their application to spread-spectrum analysis. In: 7th Information Hiding Workshop, IH05. Lecture Notes in Computer Science, Barcelona, Spain, Springer Verlag (2005)
2. Cayre, F., Fontaine, C., Furon, T.: Watermarking security part I: Theory. In: Proceedings of SPIE, Security, Steganography and Watermarking of Multimedia Contents VII. Volume 5681., San Jose, USA (2005)
3. Pérez-Freire, L., Pérez-González, F., Furon, T.: On achievable security levels for lattice data hiding in the Known Message Attack scenario. In: 8th ACM Multimedia and Security Workshop, Geneva, Switzerland (2006) 68–79 Accepted.
4. Doërr, G., Dugelay, J.L.: Security pitfalls of frame-by-frame approaches to video watermarking. IEEE Transactions on Signal Processing (formerly IEEE Transactions on Acoustics, Speech, and Signal Processing) **52** (2004)
5. Cayre, F., Fontaine, C., Furon, T.: Watermarking security part II: Practice. In: Proceedings of SPIE, Security, Steganography and Watermarking of Multimedia Contents VII. Volume 5681., San Jose, USA (2005)

6. Bas, P., Hurri, J.: Vulnerability of dm watermarking of non-iid host signals to attacks utilising the statistics of independent components. IEE proceeding, transaction on information security **153** (2006) 127–139
7. Miller, M.L., Doërr, G.J., Cox, I.J.: Applying informed coding and embedding to design a robust, high capacity watermark. IEEE Trans. on Image Processing **6**(13) (2004) 791–807
8. Costa, M.H.M.: Writing on dirty paper. IEEE Transactions on Information Theory **29**(3) (1983) 439
9. Viterbi, A.J.: CDMA: Principles of Spread Spectrum Communication. Addison-Wesley, pub-AW:adr (1995)
10. Koval, O., Voloshynovskiy, S., Deguillaume, F., Pérez-González, F., Pun, T.: Worst case additive attack against quantization-based data-hiding methods. In: Proceedings of SPIE, Security, Steganography and Watermarking of Multimedia Contents VII, San Jose, USA (2005)
11. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In LeCam, L.M., Neyman, J., eds.: Proc. of the 5th Berkeley Symp. on Mathematics Statistics and Probability. (1967)
12. He, J., Lan, M., Tan, C.L., Sung, S.Y., Low, H.B.: Initialization of cluster refinement algorithms: a review and comparative study. In: Proceedings of IEEE International Joint Conference on Neural Networks. (2004) 25–29
13. Kohonen, T.: Improved versions of learning vector quantization. In: IJCNN90. (1990) 545–550
14. Wang, C., Doërr, G., Cox, I.J.: Trellis coded modulation to improve dirty paper trellis watermarking. In: Proc. SPIE. (2007)
15. Ward, Jr., J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58** (1963) 236–244
16. Bas, P., Cayre, F.: Achieving subspace or key security for woa using natural or circular watermarking. In: ACM Multimedia and Security Workshop, Geneva, Switzerland (2006)