On the Distinguishability of Distance-Bounded Permutations in Ordered Channels

Juan M. Estevez Tapiador, *Member, IEEE*, Julio C. Hernandez-Castro, Almudena Alcaide, and Arturo Ribagorda

***Abstract*—Ordered channels, such as those provided by Internet protocol and transmission control protocol protocols, rely on sequence numbers to recover from packet reordering due to network dynamics. The existence of covert channels in any ordered channel is a well-known fact: Two parties can reorder the elements (packets) to be sent according to some predefined code. Schemes based on distance-bounded permutations have been proposed for steganographic communication with the aim of keeping and controling the increase of latency due to reordering. In this paper, we demonstrate that distance-bounded permutations are highly anomalous from a metric point of view. Our analysis is based on the study of the distribution of distances between normal permutations generated by the channel, and those produced when embedding hidden information. We provide results for four different distances: Kendall's τ, Spearman's ρ, Spearman's footrule, and Levenshtein's distance (which is equivalent to Ulam's distance for permutations). In all cases, it is shown how sequences with hidden information can be separated from the normal ones. As a result, very accurate and efficient distinguishers can be easily constructed. Finally, we study the detection capabilities of the associated detectors through a receiver operating characteristic analysis.**

***Index Terms*—Distances on permutations, ordered channels, steganalysis, steganography.**

I. Introduction

THE different ways to arrange objects in a set offers a good opportunity for steganography: There are $n!$ different permutations of n objects so a given arrangement has an entropy of $\lfloor \log_2(n!) \rfloor$ bits. Let us assume that a number of parties share a channel in which information elements (e.g., packets) are sent one at a time, but due to the nature of the channel (e.g., routing), the sequence can be sporadically reordered. In this scenario, two stegoplayers can carefully reorder the elements to be sent according to some predefined code, thus creating a covert channel.

Permutation-based steganography has been mainly motivated for its use in communication networks. In many cases, reordering does not affect the reliability of the communication since many protocols include a unique sequence number in each packet, so the original stream can be trivially recovered. Girling [7] and Wolf [14] showed in the late 1980s that covert channels of this kind exist in local-area network (LAN) protocols. Subsequently, Rowland in [12] described a number of covert channels in the transmission control protocol/Internet protocol (TCP/IP) model, including some based on reordering IP packets. In a similar work, Handel and Sandford in [8] showed that many covert channels are present in the open system interconnection (OSI) model too. From this point of view, permutation-based steganography can be framed into network steganography.

Ahsan and Kundur in [1] proposed a scheme based on toral automorphisms—a chaotic mixing system that the authors employ to reorder the sequence of packets. Chakinala *et al.* provided an extension to this work in [3] by analyzing it in a suitably defined mathematical model. As a result, they provide a characterization of the channel capacity together with polynomial time encoding and decoding schemes which asymptotically achieve this maximum channel capacity.

To the best of our knowledge, no study has been carried out on the steganalysis of these schemes. In particular, it seems natural to analyze the extent to which the reordered sequences are distinguishable from the original ones. In this paper, we tackle this question.

Our analysis is based on the study of the distribution of distances between normal permutations generated by the channel, and those produced when embedding hidden information. For this, we need a distance measure suitable for permutations. Among the existing ones, we provide results for four different metrics: Kendall's τ, Spearman's ρ, Spearman's footrule, and Levenshtein's distance (also known as "edit" distance, and equivalent to Ulam's for permutations). In all cases, the experiments show that distributions of both types of sequences can be easily told apart (as will be shown, this fact remains true even in the most unfavorable situation for the attacker). This finding will allow us to construct very accurate detectors based on any of the metrics named before.

Apart from these issues regarding distinguishability, we also discuss how an important weakness of the encoding algorithm proposed in [3] can be exploited to recover the embedded information without knowing the actual parameters employed by the stegoplayers.

The rest of this paper is organized as follows. For completeness and readability, we provide a summary of Chakinala *et al.*'s model in Section II. Section III introduces some basic metrics in permutation group spaces, particularly those that will be used in this work. In Section IV, we show through a series of experiments that stegopermutations are easily discernible from nonaltered permutations, along with the weakness found in the encoding algorithm. Section V analyzes the distinguishability power of each metric by means of receiver operating characteristic (ROC) analysis. Finally, Section VI concludes this paper and discusses some practical implications.

Manuscript received January 16, 2007; revised December 19, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jessica J. Fridrich.

The authors are with the Department of Computer Science, Carlos III University of Madrid, Madrid 28911, Spain (e-mail: jestevez@inf.uc3m.es; jcesar@inf.uc3m.es; aalcaide@inf.uc3m.es; arturo@inf.uc3m.es).

Digital Object Identifier 10.1109/TIFS.2008.920724

II. Steganographic Communication Based on Distance-Bounded Permutations

In this section, we provide an overview of the steganographic techniques based on reordering an ordered set of objects. In particular, we focus on distance-bounded permutations, although the general framework for other types of permutation-based schemes is similar.

Let $\mathcal{O} = \{o_1, \ldots, o_n\}$ be a set of objects (e.g., packets) to be sent from A to B. In what follows, we will refer to objects in $\mathcal{O}$ simply by their indexes $1, \ldots, n$. Let S_n denote the symmetric group of n elements and $e = (1, \ldots, n)$ its identity. In absence of the steganographic purposes, we assume that A sends the objects to B in the natural order given by e.

The fundamental operation performed by the stegoplayers is to permute the order in which the elements are sent. So if $\pi = (\pi(1), \ldots, \pi(n))$ is a permutation selected by A, then the first object sent to B is $\pi(1)$, the second is $\pi(2)$, and so on.

As usual, we assume that the receiving end information can only be processed in the natural order. Consequently, the communication latency of any ordered channel increases if the packets are reordered. Assume, for example, the following two permutations in S_5: $\pi = (1, 3, 2, 5, 4)$ and $\sigma = (5, 2, 3, 4, 1)$. In this case, the latency introduced by σ is clearly higher since the receiver has to wait until the last packet starts processing. These artificial delays that are associated with reordering restrict the set of possible permutations for the stegoplayers. Among the proposed models, we will focus on k-buffer permuters [3].

Definition 1 (Buffer Bounded Permuter): A k-buffer permuter uses a random-access buffer of size k elements and two operations.

1) **put**: The permuter removes one element from the input stream and places it in the buffer. This operation can be performed iff the buffer is not full.
2) **remove**: The permuter removes one element from the buffer and places it in the output stream. This operation can be performed iff the buffer is not empty.

A k-buffer permutation is a permutation that is realizable by a valid sequence of put(s) and is removed by a k-buffer permuter. Notice that the identity e is the only possible 1-buffer permutation. Let $B_n^{(k)}$ denote the number of different k-buffer permutations of order n. The following lemma (see [3]) characterizes this number.

Lemma 1: $B_n^{(k)} = k^{n-k}k!$ if $n \geqslant k$ and $B_n^{(k)} = n!$ if $n < k$.

Chakinala *et al.* propose in [3] a very efficient algorithm to encode any number between 0 and $B_n^{(k)}$ into a k-buffer permutation. Generally, this number may be seen as an index to a shared code (i.e., a table of predefined messages) between stegoplayers, although this is not a restriction of the scheme. A description of the procedure is shown in Fig. 1. The authors suggest that permuters should be implemented in hardware, a fact that can considerably increase their performance.

Merely as an example, Table I illustrates the 32 different codifications produced by a 2-buffer permuter for a group of order $n = 6$. (Note: In our experimentation, we have used the set $\{0, \ldots, n-1\}$ instead of $\{1, \ldots, n\}$).

```
1   while n > 1 do
2     Fill the k-buffer with as many elements from the input as
      possible: min(n,k)
3     Sort the k-buffer
4     for i = 1 to k do
5       if x < i B_{n-1}^{(k)} then
6         Output the i-th element of the sorted buffer
7         x ← x − (i − 1) B_{n-1}^{(k)}
8         n ← n − 1
9         break
10      end if
11    end for
12  end while
13  Output the last packet left (n = 1 here)
```

Fig. 1. Algorithm to encode any $0 \leqslant x \leqslant B_n^{(k)}$ into a k-buffer permutation using n elements [3].

TABLE I
Thirty-Two Possible Codifications Performed by a 2-Buffer Permuter $n = 6$

0 ↦ (0, 1, 2, 3, 4, 5)	16 ↦ (1, 0, 2, 3, 4, 5)
1 ↦ (0, 1, 2, 3, 5, 4)	17 ↦ (1, 0, 2, 3, 5, 4)
2 ↦ (0, 1, 2, 4, 3, 5)	18 ↦ (1, 0, 2, 4, 3, 5)
3 ↦ (0, 1, 2, 4, 5, 3)	19 ↦ (1, 0, 2, 4, 5, 3)
4 ↦ (0, 1, 3, 2, 4, 5)	20 ↦ (1, 0, 3, 2, 4, 5)
5 ↦ (0, 1, 3, 2, 5, 4)	21 ↦ (1, 0, 3, 2, 5, 4)
6 ↦ (0, 1, 3, 4, 2, 5)	22 ↦ (1, 0, 3, 4, 2, 5)
7 ↦ (0, 1, 3, 4, 5, 2)	23 ↦ (1, 0, 3, 4, 5, 2)
8 ↦ (0, 2, 1, 3, 4, 5)	24 ↦ (1, 2, 0, 3, 4, 5)
9 ↦ (0, 2, 1, 3, 5, 4)	25 ↦ (1, 2, 0, 3, 5, 4)
10 ↦ (0, 2, 1, 4, 3, 5)	26 ↦ (1, 2, 0, 4, 3, 5)
11 ↦ (0, 2, 1, 4, 5, 3)	27 ↦ (1, 2, 0, 4, 5, 3)
12 ↦ (0, 2, 3, 1, 4, 5)	28 ↦ (1, 2, 3, 0, 4, 5)
13 ↦ (0, 2, 3, 1, 5, 4)	29 ↦ (1, 2, 3, 0, 5, 4)
14 ↦ (0, 2, 3, 4, 1, 5)	30 ↦ (1, 2, 3, 4, 0, 5)
15 ↦ (0, 2, 3, 4, 5, 1)	31 ↦ (1, 2, 3, 4, 5, 0)

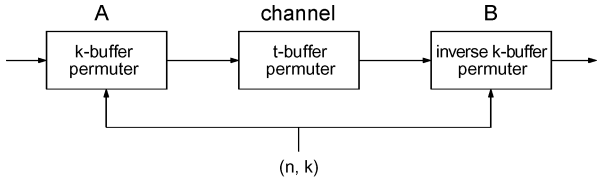


Fig. 2. Model of steganographic communication based on k-buffer permutations.

A. Model of Steganographic Communication

Fig. 2 depicts the model of steganographic communication based on reordering. Stegocommunication is modeled by using two players: stego-Alice (A) and stego-Bob (B), who work by permuting the packets before sending them. The channel is supposed to behave as a jammer, which is, in turn, another permuter. This tries to encapsulate the behavior of a real communication channel (e.g., an IP network) and the effects of attempts to disrupt the covert channel by an active warden.

Even though the channel is also modelled as a buffer bounded permutation, its permuting power should be considerably lower than that used by stegoplayers. This is a requirement in order for A and B to communicate secretly [3]. So if stegoplayers use a k-buffer permuter, the channel should behave as a t-buffer permuter with $t \ll k$. Notice that for $t = 1$, we have a nonjamming channel in which sequences are not altered at all.

III. Metrics in Permutation Group Spaces

In this section, we briefly introduce some metrics on permutations which will be extensively used throughout this work. As a general reference, the book by Kendall and Gibbons [10] provides a detailed account of the methods used here. A formal treatment of metrics on permutations can be also found in [4].

A metric on a set X is a function $d : X \times X \to \mathbb{R}$ which defines a distance between elements in X. This function is required to satisfy the following well-known axioms.

A1 Nonnegativity $d(\pi,\sigma) \geqslant 0\ \forall \pi, \sigma \in S_n$.

A2 Separation $d(\pi,\sigma) = 0$ iff $\pi = \sigma\ \forall \pi, \sigma \in S_n$.

A3 Symmetry $d(\pi,\sigma) = d(\sigma,\pi)\ \forall \pi, \sigma \in S_n$.

A4 Triangle inequality $d(\pi,\sigma) \leqslant d(\pi,\rho) + d(\rho,\sigma)\ \forall \pi, \sigma, \rho \in S_n$.

Moreover, an additional property is essential to distances in permutation group spaces.

A5 Right invariance $d(\pi,\sigma) = d(\pi \circ \rho, \sigma \circ \rho)\ \forall \pi, \sigma, \rho \in S_n$.

Most of the distances d defined over permutations are indeed metrics. Others, which formally do not satisfy some of the previous properties, are, however, useful in many applications as statistical indicators of divergences between elements.

Probably the easiest possible distance which can be defined over permutations is the generalized Hamming distance, given by

$$H(\pi,\sigma) = \text{card}\,\{i :\ \pi(i) \neq \sigma(i)\} \tag{1}$$

where $\text{card}(X)$ denotes the cardinal of the set X. Next, we introduce other well-known distances over permutations which will be used throughout this work.

Definition 2 (Spearman's ρ or Rank Correlation): Given π, $\sigma \in S_n$

$$R(\pi,\sigma) = \sqrt{\sum_{i=1}^{n} (\pi(i) - \sigma(i))^2}. \tag{2}$$

Definition 3 (Spearman's Footrule): Given $\pi, \sigma \in S_n$

$$F(\pi,\sigma) = \sum_{i=1}^{n} |\pi(i) - \sigma(i)|. \tag{3}$$

Notice that Spearman's ρ is simply the Euclidean. Its particular name comes from its use for measuring distances between rank vectors. The same is applicable to Spearman's footrule, which is simply the 1-norm distance (or Minkowski distance of order 1).

Definition 4 (Kendall's τ): Given $\pi, \sigma \in S_n$

$$T(\pi,\sigma) = \text{card}\,\{(i,j) :\ \pi(i) < \pi(j) \text{ and } \sigma(i) > \sigma(j)\}. \tag{4}$$

Kendall's distance [9] measures the minimum number of pairwise adjacent transpositions (permutations involving just swapping two elements) needed to transform π into σ. A generalization of Kendall's distance is Cayley's distance, which measures the minimum number of transpositions of any pair of elements needed to transform one permutation into the other.

A very powerful tool developed in a different context is Levenshtein's (or edit) distance [11]. Formally, it is defined over strings, which cannot be considered permutations (elements can have different lengths and repetitions are allowed).

Definition 5 (Levenshtein's (Edit) Distance): Given a finite alphabet Σ and two strings $s_1, s_2 \in \Sigma^*$, $L(s_1, s_2)$ is the minimum number of editing operations required to transform s_1 into s_2 or vice-versa, where the allowed editing operations are insertion, deletion, and replacement of a symbol inside the string.

In this formulation, Levenshtein's distance is a metric. When applied to permutations, it is strictly related to Ulam's distance

$$U(\pi,\sigma) = n - \Lambda(\pi \circ \overline{\sigma}) \quad \forall \pi, \sigma \in S_n \tag{5}$$

where $\Lambda(\alpha)$ is the length of the longest ascending subsequence of the permutation α and $\overline{\sigma}$ denotes the inverse permutation. (Recall that a sequence w is the longest common subsequence of two sequences x and y if w is a subsequence of both x and y, and its length is maximal).

Among the metrics presented before, we will use Kendall's τ, Spearman's ρ, Spearman's footrule, and Levenshtein's distance in this paper.

IV. Analysis

The purpose of this section is to show how the k-buffer permutations sent by stegoplayers (and then permuted again by the channel) can be easily distinguished from sequences not having embedded information (i.e., that have only been permuted by the channel). In such a case, effective distinguishers may be constructed in order to detect whether hidden information is being sent through the channel.

A. Experiment 1

In what follows, $\sigma = \mathsf{encode}_n^{(k)}(x,\pi)$ means that permutation $\sigma \in S_n$ is the result of encoding x into permutation $\pi \in S_n$ by using a k-buffer permuter.

Consider the following experiment: Provided values for n, k, and t, then for all $x \in \{0, 1, \ldots, B_n^{(k)}\}$.

Step 1) Compute the output generated by A

$$\theta_1 = \mathsf{encode}_n^{(k)}(x, e).$$

Step 2) Compute the actual sequence received by B

$$\theta_2 = \mathsf{encode}_n^{(t)}(r, \theta_1)$$

where r is a random number uniformly chosen in $[0, B_n^{(t)}]$. Note that this is the worst-case scenario for the steganalyst.

Step 3) Compute the sequence received by B in case A would have not embedded any information

$$\eta = \mathsf{encode}_n^{(t)}(r, e).$$

Step 4) Compute and store the distance $d(\theta_2, \eta)$ between both permutations.

For illustration purposes, we have selected $n = 10$, a six-buffer permuter for stegoplayers and a two-buffer permuter for the channel. According to Lemma 1, the number of different symbols x, which can be sent, is $B_{10}^{(6)} = 6^{10-6}6! = 933120$. Moreover, in Step 4), we have applied Levenshtein's distance,

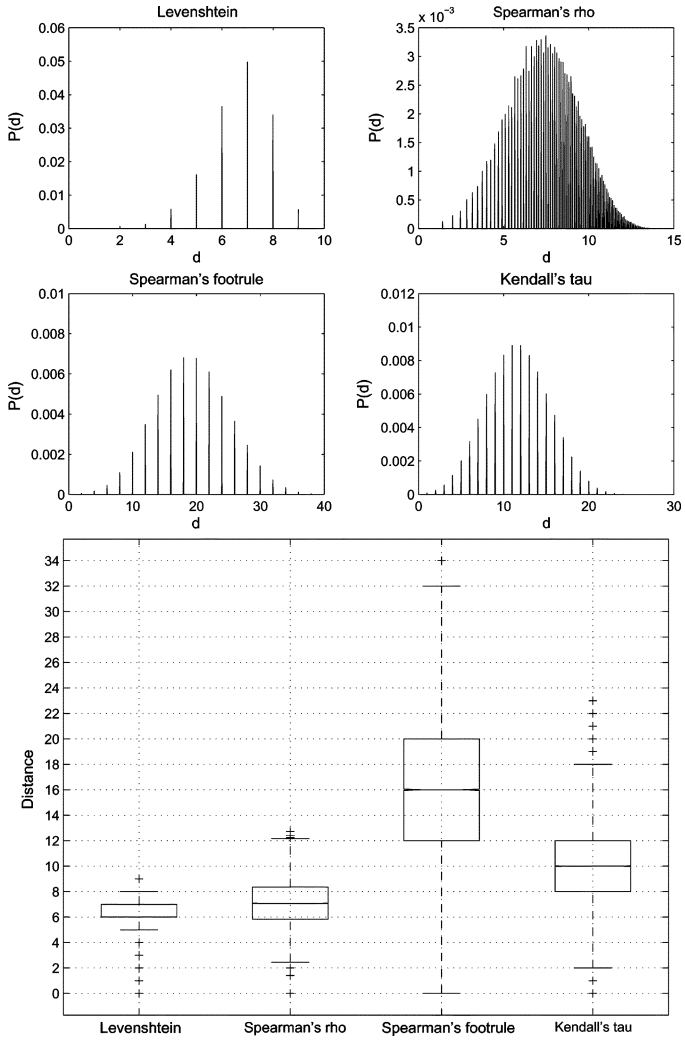


Fig. 3. Probability distribution functions of distances between channel permutations ($t = 2$) and stego permutations ($k = 6$) for $n = 10$. The bottom graph shows the boxplots for the same distributions.

Spearman's ρ, Spearman's footrule, and Kendall's τ, as defined in Section III.

Fig. 3 shows the results. The probability distribution functions for each distance reveal that stego permutations are indeed discernible from "normal" permutations. The specific features induced by each distance are better illustrated by the boxplot shown in the bottom graph. For each distance, the box has lines at the lower quartile, median, and upper quartile values. The lines extending from each box show the extent of the rest of the data. Outliers are marked by the symbol "+."

B. Experiment 2: Effect of k

Due to the very definition of the metrics, it seems reasonable to suppose that as k increases, the distance to normal permutations should also increase generally. This hypothesis has been tested by repeating the previous experiment with several values of k. Fig. 4 shows the results in the form of boxplots.

In the four cases, the distribution of distances is increasingly separated from zero as the difference between k and t becomes

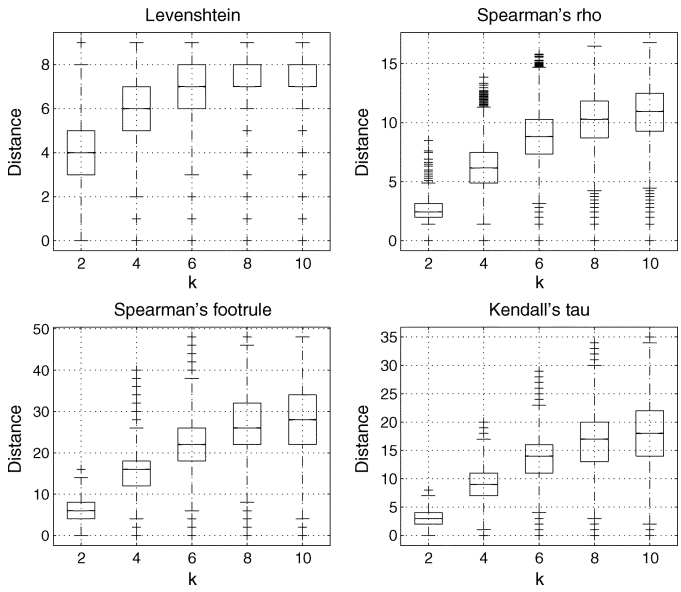


Fig. 4. Boxplots of distance distributions for $n = 10$ and different values of k. In all cases, $t = 2$.

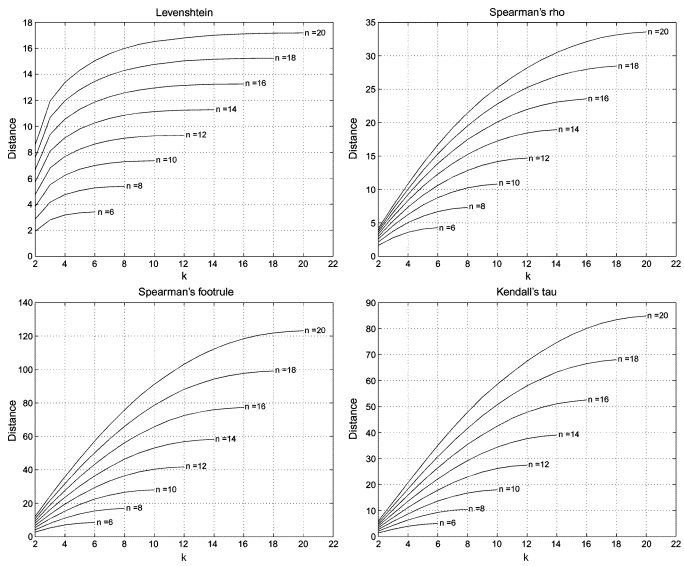


Fig. 5. Average distance between normal and stego permutations for different values of n and k. In all cases, $t = 2$.

higher. The bound for the growth is due to the fact that the experiments have been carried out with sequences of length $n = 10$, so when k approaches n, the distance reaches its maximum.

Fig. 5 shows the mean distance for different values of n and k. It seems clear that the behavior is identical regardless of the value n chosen (i.e., separability increases not only as k grows for a fixed n), but also as n becomes higher.

In summary, the two most important conclusions that can be drawn from these results are the following. First, as the power available to stegoplayers (k) increases, the deviation of the permutations with respect to the normal behavior exhibited by the channel also increases, so a more accurate detection can be performed. Second, permutations are distinguishable even in the

most unfavorable situation for the stegoplayers (i.e., $k = t$), a case in which stegocommunication is not even possible.

These two facts allow us to construct distance-based distinguishers to detect the presence of stego permutations in the channel. More details about them, along with a study of their performance, are provided in Section V.

C. On the Recoverability of the Hidden Information

Next, the property illustrates a major weakness of the encoding algorithm shown in Fig. 1:

Lemma 2 (Encoding Equivalence): Provided $k_1 \neq k_2$ and $x \leqslant B_n^{(min(k_1,k_2))}$, then

$$\mathsf{encode}_n^{(k_1)}(x,\pi) = \mathsf{encode}_n^{(k_2)}(x,\pi) \quad \forall \pi \in S_n. \tag{6}$$

A proof of this lemma is provided in the Appendix. The main implication of this property is that the exact value of k is not needed for a warden (an adversary, other than stego-Alice or stego-Bob [13]) to recover x from the observed permutation; one can decode with a higher value of k and obtain the correct hidden value. In practice, either the maximum allowed value for k (i.e., n) or an empirical overestimation of k can be used by an adversary reading the channel, and it is more recommendable to use n when possible.

We strongly believe that this is a major weakness in the encoding procedure, which is, therefore, not secure at all.

V. Distinguishability and Detection Accuracy

First, let us summarize the main conclusions of the previous analysis.

1) As the power and bandwidth (i.e., capacity) available to stegoplayers increases, the deviation of the permutations with respect to the normal behavior exhibited by the channel increases accordingly, thus facilitating more accurate detection.
2) Furthermore, permutations are distinguishable even in the most unfavorable situation for the stegoplayers (i.e., $k = t$).
3) Hidden information (x) can be recovered even without knowing the bound k secretly selected by stegoplayers if the efficient encoding algorithm proposed in [3] is used. Other encoding algorithms have been proposed, but are much less efficient (see, for example, [1]).

To measure the detection accuracy and to decide which distance measure behaves better, we have calculated the ROC curves [5] for a detector based on a distance threshold with respect to the identity permutation e. This detector simply works by checking whether the distance is higher than a given threshold

$$\text{If } d^{(n)}(\pi, e) > \theta, \text{ then alarm} \tag{7}$$

where π is the actual sequence received and n is its length.

As in previous experiments, for illustration purposes, we assume a jamming power of $t = 2$ for the channel. We have generated a random population of hidden messages x, and then encoded them into sequences using values of $k = 4, 6, 8, \ldots$ Next, we computed the distance from each permutation (both with and without hidden information) to the identity e and compared it with a threshold θ. For each value of θ, the probabilities of false alarm (FA) and correct detection (CD) are defined as

$$P(\text{FA}) = \frac{\#\text{ Normal sequences identified as stego}}{\#\text{ Total number of normal sequences}} \tag{8}$$

$$P(\text{CD}) = \frac{\#\text{ Stego sequences identified as stego}}{\#\text{ Total number of stego sequences}}. \tag{9}$$

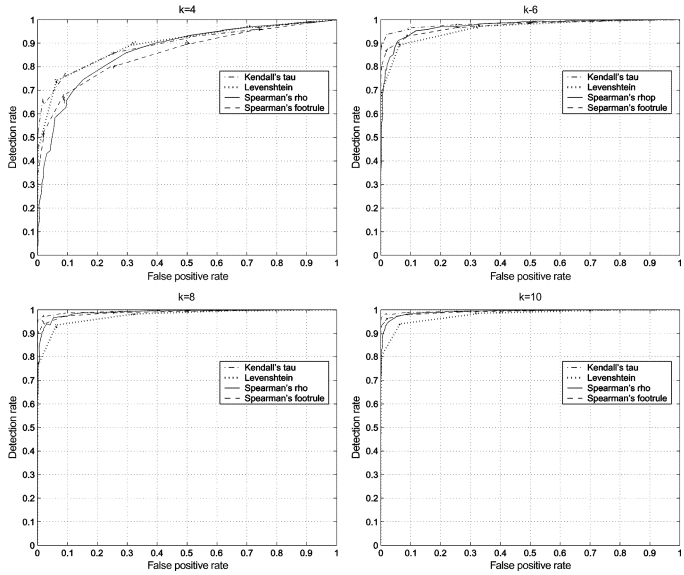


Fig. 6. ROCs of detectors based on a distance threshold. The figure compares the performance of the four distances studied in this work.

Our results are summarized in Fig. 6 for some values of k and the four distances are studied. As observed before, detection is more accurate as the difference in power between stegoplayers (k) and the channel (t) increases.

In general terms, Kendall's τ outperforms the other three distances in detection capability. As an example, for $k = 10$ and a false positive rate of 1%, the detector based on Kendall's τ produces a rate of 98.4%. In the same conditions, the detectors based on Levenshtein's, Spearman's ρ, and Spearman's footrule reach detection rates of 95.1%, 98.1%, and 97.8%, respectively.

VI. Conclusion

In this paper, we have studied the distinguishability of steganographic techniques based on distance-bounded permutations. We have demonstrated that as the power and bandwidth available to stegoplayers increases, the deviation of the permutations with respect to the normal behavior exhibited by the channel increases accordingly, thus facilitating more accurate detection. Furthermore, stegopermutations are distinguishable even in the most unfavorable situation for the stegoplayers (i.e., $k = t$).

Apart from these facts, we have shown that the encoding algorithm proposed in [3] suffers from a major weakness, for hidden information can be trivially obtained by an adversary even without knowing the bound k secretly selected by stegoplayers. Other encoding algorithms (e.g., [1]) have been proposed in this context, but they are more inefficient. As a result, it should be clear that any secret communication based on this technique can be easily disrupted and, in some cases, even the

embedded data may be recovered without knowing the key. Its use, therefore, is strongly discouraged.

Although not studied in this paper, the behavior of k-distance permuters and k-stack permuters should be, from a metric point of view, very similar to that shown by k-buffer permuters. More generally, every model using permutations (bounded or not), which are significatively different from the behavior exhibited by the channel, is likely to fail similarly. In this regard, more robust methods are required to generate indistinguishable permutations.

Our results may have some practical implications for constructing an active warden such as those described in [2]. From the point of view of an organization suspecting that some of its users may be using a covert channel of this kind (i.e., companies worried about disloyal employees leaking confidential data, or government agencies), a simple solution would be to systematically reorder every network packet flow, thus disrupting the secret communication. However, this would result in an increase of the overall latency of network applications. In scenarios where this cost should be avoided, a detector based on a distance threshold can be used at the core of an adaptive jamming policy: The more anomalous a sequence of packets seems, the heavier it is reordered. In this way, covert communications can be disrupted and, at the same time, normal communications do not suffer from a severe increase in latency. This functionality can be placed on network security devices, much in the way suggested in [6] for removing other types of covert channels usually found in network protocols.

APPENDIX
PROOF OF LEMMA 2

We first need to prove that the number of different k-buffer permutations of order n $B_n^{(k)}$ is monotonically increasing in k.

Proposition 1: For any fixed positive n, if $k_1 \leqslant k_2$, then $B_n^{(k_1)} \leqslant B_n^{(k_2)}$.

Proof: We will show that $B_n^{(k)} \leqslant B_n^{(k+1)}$ for all k, which is an equivalent definition of Proposition 1.

According to Lemma 1, two cases exist. If $k \geqslant n-1$, then $B_n^{(k)} = n!$. As this is constant for all k, the proposition holds.

In the case of $k < n-1$, by induction on k, we have:

Case $k = 1$ $B_n^{(1)} = 1^{n-1}1! = 1 \leqslant B_n^{(2)} = 2^{(n-2)}2!$.

General case. By Lemma 1, we have

$$\begin{aligned} B_n^{(k+1)} &= (k+1)^{n-k-1} \cdot (k+1)! \\ &= (k+1)^{n-k} \cdot (k+1)^{-1} \cdot (k+1) \cdot k! \\ &= (k+1)^{n-k} \cdot k!. \end{aligned} \tag{10}$$

The term $(k+1)^{n-k}$ can be expanded as

$$\begin{aligned} (k+1)^{n-k} &= \sum_{i=0}^{n-k} \binom{n-k}{i} k^{n-k-i} \\ &= k^{n-k} + \sum_{i=1}^{n-k} \binom{n-k}{i} k^{n-k-i}. \end{aligned} \tag{11}$$

Substituting (11) in (10), we have

$$\begin{aligned} B_n^{(k+1)} &= \left[k^{n-k} + \sum_{i=1}^{n-k} \binom{n-k}{i} k^{n-k-i} \right] \cdot k! \\ &= k^{n-k}k! + k! \sum_{i=1}^{n-k} \binom{n-k}{i} k^{n-k-i} \\ &= B_n^{(k)} + k! \sum_{i=1}^{n-k} \binom{n-k}{i} k^{n-k-i}. \end{aligned} \tag{12}$$

Since

$$k! \sum_{i=1}^{n-k} \binom{n-k}{i} k^{n-k-i} > 0 \quad \forall k \geqslant 0 \tag{13}$$

we have $B_n^{(k)} \leqslant B_n^{(k+1)}$. □

A. Proof of Lemma 2

Lemma 2 claims that if $k_1 \neq k_2$ and $x \leqslant B_n^{(min(k_1,k_2))}$, then

$$\mathsf{encode}_n^{(k_1)}(x,\pi) = \mathsf{encode}_n^{(k_2)}(x,\pi) \quad \forall \pi \in S_n.$$

Proof: The rationale behind the proof is the following. The encoding algorithm is deterministic in the sense that it does not select a "position" into the sequence to embed the value from that point on. Rather, it always begins to permute the first elements (see, for example, the behavior exemplified in Table I). The length to which alterations with respect to the original sequence extend depends on the value x to be embedded. Therefore, a k_1 permuter behaves identical to a k_2 permuter for those values that both permuters can codify (i.e., $x \leqslant B_n^{(min(k_1,k_2))}$). As a result, the output generated is the same in both cases.

Let P_1 and P_2 be a k_1 permuter and a k_2 permuter, respectively, with $k_1 \leqslant k_2$. Consider now a value $x \leqslant B_n^{(k_1)}$ to be embedded into a given permutation. We first prove that if P_1 passes step 5 of the encoding algorithm (see Fig. 1), then P_2 also passes it. By passing step 5, we mean that the condition is true and steps 6, 7, 8, and 9 are executed.

If P_1 passes step 5, then $x < iB_{n-1}^{(k_1)}$ for certain i. Since $B_n^{(k_1)} \leqslant B_n^{(k_2)}$ (Proposition 1), then $x < iB_{n-1}^{(k_2)}$ and P_2 passes step 5 too.

Note that now both permuters output the same element of the buffer. After doing this, P_1 and P_2 update the value of x. Let x_1 and x_2 be the updated value for permuters P_1 and P_2, respectively

$$x_1 = x - (i-1)B_{n-1}^{(k_1)} \tag{14}$$

$$x_2 = x - (i-1)B_{n-1}^{(k_2)}. \tag{15}$$

Note that after this

$$x_2 \leqslant x_1. \tag{16}$$

Consider now that P_2 passes step 5 and P_1 not. If it is the first time for P_2 to pass it (i.e., x has not been updated), the fact that P_1 does not pass it would mean that $x > iB_{n-1}^{(k_1)}$. However, this is not possible according to our assumptions on x. Likewise, if

both P_2 and P_1 have previously passed step 5, P_1 would also pass it due to relation (16).

In summary, P_1 passes step 5 of the encoding algorithm if and only if P_2 passes it too. As the output generated depends upon passing this step, both permuters produce the same sequence. □

ACKNOWLEDGMENT

The authors would like to express their gratitude to the anonymous reviewers for their insights and fruitful comments during the review process, which have greatly contributed to improve the quality of the original manuscript.

REFERENCES

[1] K. Ahsan and D. Kundur, "Practical data hiding in TCP/IP," presented at the ACM Workshop on Multimedia and Security, Juan-les-Pins, France, Dec. 6, 2002.

[2] R. J. Anderson and F. A. Petitcolas, "On the limits of steganography," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 474–481, May 1998.

[3] R. C. Chakinala, A. Kumarasubramanian, R. Manokaran, G. Noubir, C. Pandu Rangan, and R. Sundaram, "Steganographic communication in ordered channels," presented at the 8th Information Hiding Workshop, Old Town Alexandria, VA, Jul. 10–12, 2006, Lecture Notes Comput. Sci.

[4] P. Diaconis, *Group Representation in Probability and Statistics*, ser. IMS Lecture Notes Monographic Series 11. Hayward, CA: Inst. Math. Stat., 1988.

[5] J. P. Egan, *Signal Detection Theory and ROC Analysis*. Orlando, FL: Academia Press, 1975.

[6] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil, "Eliminating steganography in Internet traffic with active wardens," in *Proc. 5th Information Hiding Workshop*, Noordwijkerhout, The Netherlands, Oct. 2002, vol. 2578, Lecture Notes Comput. Sci., pp. 18–35, Springer-Verlag.

[7] C. G. Girling, "Covert channels in LAN's," *IEEE Trans. Soft. Eng.*, vol. SE-13, no. 2, pp. 292–296, Feb. 1987.

[8] T. Handel and M. Sandford, "Hiding data in the OSI model," in *First Int. Workshop on Information Hiding*, Cambridge, U.K., May/Jun. 1996, vol. 1174, Lecture Notes Comput. Sci., pp. 23–28, Springer-Verlag.

[9] M. Kendall, "A new measure of rank correlation," *Biometrica*, vol. 30, pp. 81–89, 1938.

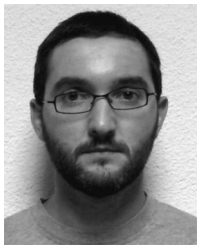
[10] M. Kendall and J. D. Gibbons, *Rank Correlation Methods*. London, U.K.: Edward Arnold, 1990.

[11] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Sov. Phys. Dokl.*, vol. 6, pp. 707–710, 1966.

[12] C. H. Rowland, "Covert channels in the TCP/IP protocol suite," *First Monday*, Jul. 1997.

[13] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Proc. Advances Cryptography:* , D. Chaum, Ed., Aug. 1983, pp. 51–67.

[14] M. Wolf, "Covert channels in LAN protocols," in *Proc. Workshop Local Area Network Security*, 1989, pp. 91–102.



Juan M. Estevez Tapiador (M'02) received the M.Sc. and Ph.D. degrees in computer science from the University of Granada, Granada, Spain, in 2000 and 2004, respectively.

Currently, he is Associate Professor at the Computer Science Department of Carlos III University of Madrid, Madrid, Spain. His research is focused on cryptography and information security, especially in formal methods applied to computer security, design, and analysis of cryptographic protocols and steganography. In these fields, he has published around 40 papers in specialized journals and conference proceedings. During 2007, he was Visiting Professor at the Department of Computer Science, University of York.

Dr. Tapiador received the Best Student Academic Award from the University of Granada in 2004. He is member of the program committee of several conferences related to information security and is Regular Referee for various journals.

Julio C. Hernandez-Castro received the B.Sc. degree in mathematics from the Universidad Complutense de Madrid, Madrid, Spain, in 1995, the M.Sc. degree in coding theory and network security from the Universidad de Valladolid in 1999, and the Ph.D. degree in computer science from Carlos III University of Madrid in 2003.

Currently, he is Associate Professor at the Computer Science Department of Carlos III University of Madrid. His interests are mainly focused in cryptology, network security, steganography, and evolutionary computation.

Almudena Alcaide received the B.Sc. degree in mathematics from the Complutense University of Madrid, Madrid, Spain, the M.Sc. degree in advanced computing from King's College of London, London, U.K., and is currently pursuing the Ph.D. degree in computer science from the Carlos III University of Madrid.

Currently, she is Assistant Professor at the Cryptography and Information Security Group of the Computer Science Department of Carlos III University of Madrid. Her work is focused on formal methods applied to the design and analysis of cryptographic protocols. Her most recent research activity is related to applying game theory results to security protocols, and how these methods can assist in developing automated protocol verification tools and techniques.

Arturo Ribagorda received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Universidad Politecnica de Madrid, Madrid, Spain, in 1978 and 1983, respectively.

Currently, he is Full Professor at Carlos III University of Madrid, Madrid, Spain, where he is also the Head of the Cryptography and Information Security Group and Director of the Computer Science Department. He is one of the pioneers of computer security in Spain, having more than 25 years of research and development experience in this field. He has authored four books and many articles in several areas of information security.

Dr. Ribagorda is a member of the program committee of several conferences related to cryptography and information security.