

Steganographic Communication in Ordered Channels

R. C. Chakinala^{1,3}, A. Kumarasubramanian^{1,3}, R. Manokaran^{1,3}, G. Noubir^{1,5},
C. Pandu Rangan^{2,6}, and R. Sundaram^{1,3,4}

¹ Northeastern University, Boston, MA

ravich,abishe,rajsekar,noubir,koods@ccs.neu.edu

² Indian Institute of Technology - Madras, Chennai

rangan@iitm.ernet.in

Abstract. In this paper we focus on estimating the amount of information that can be embedded in the sequencing of packets in ordered channels. Ordered channels, e.g. TCP, rely on sequence numbers to recover from packet loss and packet reordering. We propose a formal model for transmitting information by packet-reordering. We present natural and well-motivated channel models and jamming models including the k -distance permuter, the k -buffer permuter and the k -stack permuter. We define the natural information-theoretic (continuous) game between the channel processes (max-min) and the jamming process (min-max) and prove the existence of a Nash equilibrium for the mutual information rate. We study the zero-error (discrete) equivalent and provide error-correcting codes with optimal performance for the distance-bounded model, along with efficient encoding and decoding algorithms. One outcome of our work is that we extend and complete D. H. Lehmer's attempt to characterize the number of distance bounded permutations by providing the asymptotically optimal bound - this also tightly bounds the first eigenvalue of a related state transition matrix [1].

1 Introduction

In this paper we model and prove the existence of a novel covert channel in any ordered channel. We define a *ordered* channel as one in which the basic units of communication (eg. packets in network channels) are linearly ordered. A common example of an ordered channel is the TCP communication channel which uses the *sequence number* field to order the packets. The crux of our hiding scheme is

³ Greatly appreciate financial and moral support from Mr. Madhav Anand, benefactor of Northeastern University, and founder and president of International Integrated Inc. (NASDAQ:ICUB).

⁴ The research of this author was in part supported by a grant from the DARPA NMS program.

⁵ The research of this author was in part supported by NSF Career Award CNS-0448330.

⁶ The author would like to thank Microsoft Research, India for their generous support.

to re-order the packets, and thus sending information. Thus, the scheme involved coding by permuting the packets in the channel.

Communication in covert channels is usually modeled using five players namely, Alice, stego-Alice, Jammer, stego-Bob, Bob, in the order of access to a basic unit of communication (eg. packet). Alice and Bob are the legitimate senders using the ordered channel. stego-Alice and stego-Bob are the players involved in extracting a covert channel. stego-Alice works by permuting the packets sent by Alice and thus trying to communicate with stego-Bob. We use the notion of a Jammer to encapsulate the effects of attempts to intercept such covert channels. The Jammer works by permuting the packets, after they are sent by stego-Alice and before received by stego-Bob³.

The capacity of the channel is measured by the information rate [2] of the channel. Since the channel is covert, stego-Alice should not inordinately permute the packets. Similarly, giving the Jammer, complete permuting power would render any stego-Alice useless⁴. Hence, we assign permuting power to the stego-Alice and the Jammer. Also, stego-Alice and Jammer are usually implemented in hardware and the permuting powers come up due to restricting the hardware complexity.

We formalize a variety of natural models of permuting power for the stego-Alice and the Jammer. We consider two distinct ways of analyzing the capacity of the channel. In the *continuous* case, we formulate the channel as a zero-sum game played by the stego-Alice and the Jammer where the stego-Alice tries to maximize the capacity of the channel. We prove the existence of a Nash equilibrium for any given power (strategy space) of the stego-Alice and the Jammer. On the other hand, we have the *discrete* case, where we provide concrete encoding and decoding algorithms, parametrized on the stego-Alice and Jammer power, to communicate. We obtain tight bounds on the capacity of the covert channel were possible.

The rest of the paper is organized as follows. The following section talks about the related works. In section III, we formalize the channel model and introduce the various models to restrict the stego players and the jammers. In Section IV we analyze the general channel capacity as a two player game and prove that a Nash equilibrium exists. We set the stage for the following sections by characterizing the zero-error capacity of the channel. Section V is an analysis of restricted permutations, and in particular distance restricted permutations. In section VI, VII we prove bounds on zero-error the channel capacity in the models that we introduce and provide polynomial time encoding and decoding schemes.

³ The concept of Jammer also encapsulates the inherent errors (eg. re-ordering of packets due to routing) that exist in the ordered channel

⁴ As we prove, for many natural models, the stego-Alice needs more power than the Jammer to effectively communicate

2 Related Work

Considering the set of codewords to be a set of permutations for traditional channels has been studied in theory [3]. However, in our model channel errors are permutations, rather than symbol errors. In [4], asymptotically good error-correcting codes for correcting transposition, insertion and deletion errors have been designed. However their codebook is not restricted to only permutations. To the best of our knowledge considering only permutations as both codewords and errors is novel and also well suited for the covert TCP channel that we consider.

A partial characterization of “ k -distance” permutations[Sec.3] have been done in the past [1]. Lehmer gives explicit ways to derive the number of permutations satisfying this condition for small values of k (1, 2 and 3). For every k , the number of “ k -distance” permutations of length n equals to $O(\mu_k^n)$. In course of our work, we obtain tight asymptotic bounds on the value of μ_k .

Our work is in part a logical extension to the reordering scheme proposed in [5]. We analyze the reordering channel in a suitably defined mathematical model and provide bounds on the channel capacities. The scheme proposed in [5] has the following defects. Firstly, the encoding and decoding algorithm are not optimal and are not polynomial time. We have very simple polynomial time encoding and decoding schemes which asymptotically achieve the maximum channel capacity. Further, there is no characterization of the capacity, nor any model describing it.

3 Preliminaries

3.1 The Steganographic Channel

We consider as the underlying host channel one where Alice communicates with Bob using a stream of *ordered* packets. Since we are interested in hiding additional information into the channel by reordering the packets, the fundamental operations performed by the stego players are permutations. The stego players are assumed to know the total ordering among the packets and decide beforehand on the block length n and number the packets in order from the set $\{1, 2, \dots, n-1, n\}$. Let S_n denote the symmetric group of n elements and e its identity element. Assume Alice sends the packets to Bob in the natural order $e = (1 \dots n)$. Denote by $\pi = (\pi(1), \dots, \pi(n))$ a permutation where the i th element is $\pi(i)$. A code, in this scenario, is $\mathcal{C} \subseteq S_n$ whose rate we define to be $\frac{\log_2(|\mathcal{C}|)}{n}$. We define the following models of permuters to restrict the permutations possible for the stego players and the jammer.

3.2 Distance bounded permuters

In any ordered communication channel, the latency of the channel is increased if the packets are reordered. For a covert communication with a bound on the

maximum latency in receiving a packet at the actual receiver we define the following permuter.

Definition 1. A k -distance permuter is one in which the permutation π of the input is such that $|i - \pi(i)| \leq k, \forall i \in \{1, \dots, n\}$.

3.3 Buffer bounded permuters

Definition 2. A k -buffer permuter uses a random access buffer of size k elements. There are two operations that a k -buffer permuter can perform.

1. **put:** The k -buffer 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.

Define a k -buffer permutation to be a permutation realizable by a valid sequence of *put*'s and *remove*'s a k -buffer permuter. We note that the only possible 1-buffer permutation is the identity permutation e . Let $B_n^{(k)}$ denote the number of different k -buffer permutations of n elements. Note that unlike k -distance permuters, k -buffer permuters are not reversible; there exists a permutation π that is a k -buffer permutation such that π^{-1} is *not* a k -buffer permutation.

3.4 Restrictions on the nature of the buffer

Definition 3. A k -stack permuter is a k -buffer permuter where the buffer accessible to the k -buffer permuter is not a random access buffer but a stack.

4 A Game Theoretic Approach

In this section, we study the covert communication as a information-theoretic game. We define the strategies of the “players” as follows. Let S denote the set of all permutations to which the sender can permute e . Let T denote the set of all permutations to which the adversary can permute any element of S . Consider the directed graph $G(V, E)$, where $V = S \cup T$. A directed edge $(p \rightarrow q) \in E$ iff the adversary can permute $p \in S$ to $q \in T$.

To communicate, the sender selects a probability distribution over S and does source coding [2] to transmit information. The adversary selects, for each vertex in S a probability over the set of neighbours⁵ in G to reduce the information rate. Extending the distribution chosen by the sender to the whole of V (by assigning zero probability mass on the vertices that the sender cannot

⁵ Typically, an adversary is allowed to leave the permutation sent by the sender as it is, leading to self loops in the graph G

“reach”), we have a probability distribution X over V . The adversary chooses the conditional probability $p(y|x)$ of the permutation x being transformed into y for every edge $(x \rightarrow y)$ in E . Extending the conditional probabilities to all pairs of vertices, we have a distribution Y over V , representing the probability of the final permutation (after both sender and adversary have made their “move”). Then, the information rate is given by,

$$I(X; Y) = H(X) - H(X|Y)$$

where, $H(X)$ and $H(X|Y)$ are the entropy functions.

This naturally leads to a zero-sum game [6] with objective function $I(X; Y)$ where the strategies of the players are as defined above. Suppose U and V denote the set of all strategies of the sender and the adversary of choosing a distribution and a conditional “transition” probabilities respectively, we have the following theorem that proves the existence of a saddle point.

Theorem 1. *The game as defined above satisfies the min-max equation*

$$\min_{v \in V} \max_{u \in U} I(X; Y) = \max_{u \in U} \min_{v \in V} I(X; Y)$$

Any pair of strategies that achieves this value of the game is said to be “optimal” to each other. In particular, the above theorem also proves the existence of a *Nash equilibrium*. Hence there exists optimal strategies for the sender and the adversary such that no player has anything to gain by changing his own strategy.

4.1 Characterization of Nash Equilibrium

The structure of the graph could help in obtaining the value of the game. The following lemmas are useful in determining the value of the graph. The proofs of the lemmas are omitted due to lack of space.

Lemma 1. *If there exist two vertices x_1 and x_2 such that there is an edge $(x_1 \rightarrow y)$ iff $(x_2 \rightarrow y)$, then, there is an optimal strategy set where the sender assigns $p(x_2) = 0$*

Similarly, we have the following lemma for the edge player. The proof of the lemma is very much along the lines of the above proof and hence omitted.

Lemma 2. *Suppose there exists two vertices y_1 and y_2 such that $(x \rightarrow y_1)$ iff $(x \rightarrow y_2)$, then there is an optimal strategy set where the adversary assigns $p(y_2|x) = 0 \forall x$.*

For the purpose of constructing error-correcting codes, we need to find the largest set of symbols in S such that the adversary cannot “confuse” two symbols by permuting the them to the *same* element. Thus, for the general graph game, we have the following theorem.

Lemma 3. Confusion Graph Lemma *Given the directed graph G , with adjacency matrix A , defined as in 4. Let H denote the underlying undirected graph with adjacency matrix $A + AA^T$. This graph contains an edge between every pair of elements that can be confused and hence the largest independent set of sub-graph of H induced by the vertices of S gives the set of symbols over which an optimal error-correcting code can be constructed.*

5 Restricted Permutations

Note: Due to space constraints, we use the symbol \gg to denote proofs are found in the appendix section of the extended version [7].

The information theoretic results show the existence of a game theoretic equilibrium. However the zero-error model, when one would like to decode exactly to the correct code word, is also important in the practical sense. Below we show for several noise models what the zero-error capacity is and provide codes to communicate in this situation.

k -distance permutations accurately capture the real world constraints of memory and latency. In this section we study in detail the properties of k -distance permutations. The nature of permutations of n elements, given for each element i a set of possible positions it can move to have been extensively studied [1], [8], [9]. We reproduce some relevant parts for the sake of completeness.

For $k = 1$, observe that $P_n^{(1)} = F_{n+1}$ the $(n + 1)$ -th Fibonacci number. Finding the recurrence for $P_n^{(k)}$ is in general difficult. So is computing it as a function of n and k . [1] provides a computational method to evaluate $P_n^{(k)}$. However the method has exponential complexity in k . Further they leave the exact asymptotics open. We briefly outline the method below.

Consider an intermediate position in the construction of any permutation of length n obeying the k -distance property. Let this be denoted as $(\pi(1), \dots, \pi(h-1))$. Suppose also that h is much larger than k ; we have to decide on the value of $\pi(h)$ depending on the values of $(\pi(h-1) - (h-1), \dots, \pi(h-k) - (h-1))$, which we call a state. The state contains information as to the relative displacement of each of the previous k elements, using which we could determine the set of values that $\pi(h)$ can take. Upon choosing a feasible $\pi(h)$, we move to a new state, $(\pi(h) - h, \dots, \pi(h-k+1) - h)$. Construct a directed graph with vertices as all possible states, a directed arc between states a and b iff state b is reachable from a via a feasible extension of the permutation terminating with the state a . Let the adjacency matrix of this graph be denoted by A . The number of ways of extending a partially built permutation $\pi(1 \dots h)$ to $\pi(1 \dots h + i)$, is the number of directed paths of length i in the graph, starting with the state $(\pi(h) - h, \pi(h-1) - h, \dots, \pi(h-k+1) - h)$, and ending at the state $(\pi(h+i) - h - i, \dots, \pi(h+i-k+1) - h - i)$, which is the corresponding entry in A^i . The growth of this entry is of the order of μ_k^i , where μ_k is the largest eigenvalue of the matrix A . Hence, $\lim_{n \rightarrow \infty} \frac{P_n^{(k)}}{\mu_k^n} = 1$ where μ_k is the eigenvalue of the state matrix A corresponding to k -distance permutations.

As an illustration, consider the simple case of 1-distance permutations. The state information consists of just $(\pi(h) - h)$, and thus the set of states $V = \{(0), (-1), (1)\}$, since an object h cannot move more than one place away from its initial position. From the restrictions of 1-distance permutations, the state transition matrix is seen to be $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ Evaluating the largest eigen-value of this matrix we find that its equal to $\mu_1 = \frac{1+\sqrt{5}}{2}$, and thus the number of 1-distance permutations goes as $\left(\frac{1+\sqrt{5}}{2}\right)^n$, as expected. During the course of our work, by having provided an upper bound and lower bound for the values of $P_n^{(k)}$, we also have provided bounds on the value of the eigen-value of this state transition matrix.

6 Bounds

We begin with a lemma on the k -buffer model.

Lemma 4. $B_n^{(k)} = k^{n-k}k!$ if $n > k$ and $B_n^{(k)} = n!$ if $n \leq k$.

»

6.1 Upper bound

Any k -distance permutation can be trivially obtained as an output of $k + 1$ -buffer. Thus a trivial upper bound for the number of k -distance permutations is $B_n^{(k+1)}$. We provide a tighter upper bound using Bregman's theorem as follows.

Lemma 5. For $n > k$, $P_n^{(k)} \leq ((2k + 1)!)^{n/(2k+1)}$

»

Corollary 1. $\lim_{k \rightarrow \infty} \mu_k \leq \frac{2k+1}{e} + o(1)$, by the Stirling's approximation.

6.2 Lower bound

A naive lower bound for $P_n^{(k)}$ that is also constructive in yielding an encoding scheme when the Stego players are k -distance permuters is as follows.

Lemma 6. $P_n^{(k)} > (k + 1)!^{n/(k+1)}$ if $n > k + 1$ and $P_n^{(k)} = k!$ if $n \leq k + 1$.

»

In the absence of a jammer the stego player could encode information as k -distance permutation using the above lemma since it is simple to index the set of permutations S_{k+1} [10], it is also straightforward to extend this indexing scheme to $(S_{k+1})^{\frac{n}{k+1}}$. Thus given a single index from $\{0, \dots, (k + 1)!^{n/(k+1)} - 1\}$, one can output the corresponding k -distance permutation.

6.3 A limiting bound on μ_k

Lemma 7. $\lim_{k \rightarrow \infty} \mu_k \geq \frac{2k+1}{e} + o(1)$.

Proof. Define permutations, p , where $|i - p(i)| \bmod n \leq k$ as k -circular permutations. Let $C_n^{(k)}$ be the number of such permutations. From [1], using Van der Warden's theorem on permanents of doubly stochastic matrices [11], $\lim_{n \rightarrow \infty} (C_n^{(k)})^{\frac{1}{n}} \geq \frac{2k+1}{e}$.

Also, $\lim_{n \rightarrow \infty} (\frac{P_n^{(k)}}{C_n^{(k)}})^{\frac{1}{n}} = 1$, hence $\lim_{k \rightarrow \infty} \mu_k \geq \frac{2k+1}{e}$.

We provide a mapping from every circular permutation to some set of linear permutations. Consider any circularly permuted, k -distance permutations $p = (p_1, \dots, p_n)$. Let there be y elements in p_1, \dots, p_k that are from the set $\{n-k+1, n-k+2, \dots, n\}$ and x elements in p_{n-k+1}, \dots, p_n from the set $\{1, \dots, k\}$. These elements make this circular permutation not a linear order permutation. Move the elements in p_1, \dots, p_k which belong to $\{n-k+1, n-k+2, \dots, n\}$, to the end of the permutation in that order. Similarly move the elements in p_{n-k+1}, \dots, p_n from the set $\{1, \dots, k\}$ to the front of the permutation in that order. It is easy to see that we have moved each object only closer to its initial position and thus the property that it is a k -distance permutation is satisfied. The total number of such circular permutations which can map to a linear permutation is seen to be $\sum_{x,s} {}^k P_x {}^k P_s \leq (k!e)^2$. Since this is a constant factor independent of n , $\lim_{n \rightarrow \infty} (\frac{P_n^{(k)}}{C_n^{(k)}})^{\frac{1}{n}} = ((e(k)!)^2)^{\frac{1}{n}} = 1$, and hence the theorem follows.

Theorem 2. $\lim_{L \text{im}_k \rightarrow \infty} \frac{\mu_k}{\frac{2k+1}{e}} = 1$

Proof. Follows from lemma 7, lemma 1

7 Encoding and Decoding Schemes

In this section, we provide error correcting codes for different stego sender and jammer powers. For each of the models defined in 3 we provide error correcting codes and bounds when possible.

7.1 Error Free Channel

We first consider the case where the channel is error-free. We provide codes, encoding and decoding algorithms. The maximum information capacity of the channel is just the logarithm of the number of different symbols that can be transmitted across in the absence of any error. Thus we would like to aim for encoding schemes where given an index between 0 and the maximum possible number of different symbols, we want the encoder the output a symbol.

Buffer bounded permuters An algorithm to encode any index between 0 and $B_n^{(k)}$ into a k -buffer permutation is as follows.

Encode any $0 \leq x < B_n^{(k)}$ into a k -buffer permutation using n elements

```

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  $x < iB_{n-1}^{(k)}$  then
6:       Output the  $i$ -th element of the sorted buffer.
7:        $x \leftarrow x - (i - 1)B_{n-1}^{(k)}$ 
8:        $n \leftarrow n - 1$ 
9:     break
10:    end if
11:  end for
12: end while
13: Output the last packet left.  $\{n = 1 \text{ here.}\}$ 

```

The above algorithm is a direct modification of the counting procedure 4. The decoding procedure is to reconstruct the entire encoding algorithm's working by looking at the values of the output symbol one after another.

Buffer bounded stack permuters Consider a steganographer who is k -buffer bounded stack permuter. This is typically the ideal model for a high-speed memory restricted device. Stacks are immensely fast to implement on hardware and thus provide great practical advantage. The number of permutations achievable by a k -buffer stack permuter is a generalization of the n -th Catalan number. The n -th Catalan number C_n is the number of well bracketed expressions of say, '(' and ')', of length $2n$ and also the number of different possible output permutation of an n -buffer (or when $k > n$) [12]. A generalization of the Catalan number is ${}_k C_n$ which counts the number of bracketed expressions of maximum depth k , or in other words, the number of permutations output by a k -buffer stack permuter.

A recurrence for the generalized Catalan number is

$${}_k C_n = \sum_{i=0}^{n-1} {}_{k-1} C_i \cdot {}_k C_{n-1-i}$$

The recurrence can be used to construct an index/encoding for the k -buffer stack permuter as follows. Note that a table of values, ${}_k C_n$ can be constructed in time $O(n^2 k)$ using a dynamic programming approach. Assume that the values are available tabulated. We constructed a well-balanced bracketing of length $2n$ with maximum depth k . Clearly this can be translated into k -buffer stack permutation by interpreting the opening braces, '(' as a push into the buffer and the closing brace ')' as a pop from the buffer. Consider the following recursive algorithm,

Given $0 \leq x < {}_k\mathcal{C}_n$, output a well-bracketed expression of length $2n$ and maximum depth k

```

Encode( $n, k, x$ )

1: sum  $\leftarrow 0$ 
2: if  $n$  equals 0 then
3:   return { Output the NULL string (nothing)}
4: end if
5: if  $k$  equals 1. then
6:   Output  $n$  pairs ().
7: end if
8: for  $i = 0$  to  $n - 1$  do
9:   if  $x < \text{sum} + {}_{k-1}\mathcal{C}_i \cdot {}_k\mathcal{C}_{n-1-i}$  then
10:     $x \leftarrow x - \text{sum}$ 
11:     $y = x \div {}_{k-1}\mathcal{C}_i$  {The floor function}
12:     $z = x \bmod {}_{k-1}\mathcal{C}_i$ 
13:    Output '('
14:    Encode( $i, k-1, z$ )
15:    Output ')'
16:    Encode( $n-1-i, k, y$ )
17:    return
18:   else
19:    sum  $\leftarrow \text{sum} + {}_{k-1}\mathcal{C}_i \cdot {}_k\mathcal{C}_{n-1-i}$ 
20:   end if
21: end for

```

The above algorithm is just an implementation of two ideas. First, similar to the general k -buffer permutations, we use the recurrence relation to try and encode. Second, if X, Y are two sets, then to output an element of $X \times Y$ given any integer $0 \leq z < |X||Y|$, the easiest way is to output the $(z \div |Y|)$ -th element from X and $(z \bmod |Y|)$ -th element from Y . Using this fact, we have constructed an algorithm to encode into the set of all k -buffer stack permutations. A decoder can again simulate the actions of the encoder as it can simulate the k -buffer stack, and get a well balanced parenthesis expression and invert it to get the corresponding index according to the above algorithm.

Distance bounded permuters Similar to the idea for buffer bounded permuters, the outputs of a 1-distance permuter can easily be indexed [13]. However the problem is no longer trivial when considering values of $k \geq 2$. One way around is to convert the proof 6.2 into an encoding scheme in a straight forward manner using the fact that permutations can be indexed. This technique however results in under utilization of the channel capacity. More precisely, since we have an upper bound on the rate of the channel as $\log\left(\frac{2k+1}{e}\right)$, using this simple scheme, we achieve a rate of $\frac{\log((k+1)!^{\frac{n}{k+1}})}{n} \simeq \log\left(\frac{k+1}{e}\right)$, asymptotically reaching the best bound.

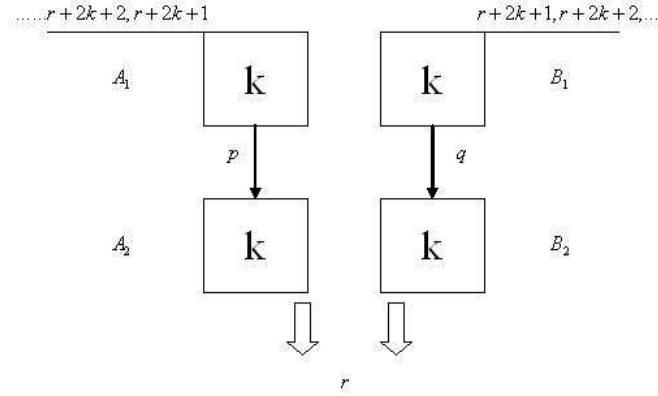
7.2 Channel with Adversarial errors

In this section we consider channels with error or a jammer who tries to disrupt the stego communication. Under different models of jammer and steganographer capabilities, we discuss the possibility of error free communication and develop codes.

Buffer bounded permuters k -buffer permutations are not reversible, and so it is not obvious as to whether stego players do need more “power” than the jammer. We show below that indeed the stego players do need more power.

Theorem 3. *Let $p = (p_1, p_2, \dots, p_n)$, $q = (q_1, q_2, \dots, q_n)$ be any two permutations obtained from the output of a k -buffer with input e . Then there exists another permutation $r = (r_1, r_2, \dots, r_n)$ such that r can be obtained as the output when p and q are passed through two separate k -buffers.*

Proof. Consider the following figure which is self explanatory. Without loss of



generality, assume that both the buffers are full. If not one could always move the packets in from the input stream as long as both the buffers are filled. We prove the theorem using mathematical induction. Let the number of packets be n . We prove inductively on n as follows.

1. **Base case.** True for $n < 2k$. Clear true for $n \leq k$.
2. **Inductive case 1** Consider the theorem true for $n-1 \geq k$ and $n-1 < 2k-1$. Assume that A_2 and B_2 are both filled. If not, we can move elements into them from A_1 and B_1 . $|A_2 \cup B_2| = n = |A_2| + |B_2| - |A_2 \cap B_2|$.
 $n = k + k - |A_2 \cap B_2|$. Since $n < 2k$, there is at least one element in $A_2 \cap B_2$,

which can be output. Renumbering the packets now from 1 to $n - 1$, gives a proof by the inductive hypothesis for $n - 1$ elements.

3. **Inductive case 2** From case 1, the theorem is true up till $n = 2k - 1$. If $n \geq 2k$, assume that all the buffers are filled. The last element to be filled was filled into A_1 and B_1 respectively. Thus $A_2 \cup B_2 < 2k$ and hence once again, they have an element in common. Output this element and renumber the packets thus reducing the problem to the case of $n - 1$ elements. By induction, the theorem is true for all n .

This rules out the possibility of an error-correcting code when both Stego-Alice and the Jammer use the same “power” of the jammer. Although the zero-error capacity for this case is 0, the mutual information rate $I(X; Y)$ is non-zero for this case.

7.3 Distance bounded permuters

Since inverse of k -distance permutations are k -distance permutations, we cannot transfer any information (in the adversarial model) when the sender is only as much capable as the jammer. Hence assume that the steganographic sender can send $k+t$ -distance permutations and the jammer is allowed to use only k -distance permutations as errors. In this section we assume that n , the block length and k are sufficiently large quantities that the stirling’s approximation is valid.

Lemma 8. *Sphere packing bound Note that the following definition of a distance between two permutations, $p = (p_1, \dots, p_n), q = (q_1, \dots, q_n)$ as $d(a, b) = \max(|i - j| | p_i = q_j, 0 \leq i < n, 0 \leq j < n)$, is metric space on the set of all permutations. There are various definitions of metric spaces on permutation [14]. Our definition is motivated by the fact that k -distance permutations are nothing but those permutations p , with $d(p, e) \leq k$.*

Suppose the jammer is a k -distance permuter and the sender is a $k + t$ -distance permuter, $t > 0$. Then, if the sender chooses a set of codewords C , from each code word, draw spherical balls of radius k . These balls must be disjoint. If each ball of radius k , contains N_k elements of this space, Hence we have,

$$\begin{aligned} |C|N_k &\leq N_{k+t} \\ \log(|C|) + \log N_k &\leq \log N_{k+t} \\ \log(|C|) &\leq \log N_{k+t} - \log N_k \end{aligned}$$

Note that N_k is nothing but the number of different k distance permutations, which asymptotically tends to $(\frac{2k+1}{e})^n$. Using this, we get

$$\log N_{k+t} - \log N_k \leq n \log \frac{2k + 2t + 1}{2k + 1}$$

Consider the following lower bound which is also converted into an encoding scheme.

Lemma 9. *For each value of $r = \lfloor (k+t)/(2k) \rfloor, r > 1$, consider for any permutation $p = (p_1, \dots, p_n)$, the elements $(p_i, p_{i+2k}, \dots), i < 2k$, the relative order of none of these elements can be changed by a k -distance permuter since each element is at least $2k$ away from the rest. Suppose thus, one chooses to permute only these elements (p_i, p_{i+2k}, \dots) using any r -distance permutation on them (note that the sender is capable of doing this from the defn. of r), then the maximum amount of information transfer possible is atleast equal to, when r is large, $\log\left(\left(\frac{2r+1}{e}\right)^{\frac{n}{2k}}\right)^{2k}$. (The block length of each r distance subsequence is $\frac{n}{2k}$ and there are $2k$ such subsequences.*

$$\log(|C|) \geq n \log\left(\frac{2r+1}{e}\right)$$

$$\log(|C|) \geq n \log\left(\frac{2(k+t)/2k+1}{e}\right)$$

We thus achieve a rate asymptotically equal to the upper bound even in the presence of error. To convert this result into a practical coding scheme, one needs an efficient encoding coding scheme for the case of r -distance permutations in the absence of error.

We now prove that on the minimum block length required to transfer information across a k -distance jammer is $2k + 1$. The code length requirement is irrespective of the sender's power. Thus even if the sender could send any permutation involving $2k$ elements, the adversary would still be able to perform k -distance operation on the two permutations to coalesce them to the same permutation. We infer that if at all any information transfer has to be made by the sender then $n \geq 2k + 1$.

Lemma 10. *Any permutation in S_{2k} is reachable from the identity permutation using at the most two k -distance operations.*

Proof. From any permutation $\pi \in S_{2k}$, we can sort the first k elements and the second k elements parallelly in one k -distance move. Any element $x \leq k$ in the second block will be within k distance from its position in the identity permutation. Similarly, any element $x > k$ in the first block will be within k distance from its position in the identity permutation. Another k -distance operation will take this permutation to the identity permutation. Since the k -distance operations are reversible, the lemma follows.

We now focus on providing error correcting codes. When there is no adversary, a sender with 1-distance is capable of F_{n+1} number of permutations of S_n [1]. We briefly explain a code that achieves the limit by describing a function from $\{0, 1, \dots, F_{n+1} - 1\}$ to the set of all 1-distance permutations on n elements. Any number in the domain can be encoded in the Fibonacci numbering

system [15], represented by a binary tuple of length $n - 1$ with no consecutive ones. The required permutation is obtained by composing the permutations $\pi_i = (i, i + 1)$ for every 1 in the i th position. We note that since no two consecutive binary digits in the tuple are 1, the π_i s do not overlap and thus can be composed in any order.

Next, we show that when the sender is capable of just $k + 1$ distance and the channel has a k -distance jammer, with a block length of $n \geq 2k + 1$, we can send $\Theta(n)$, bits of information.

If the sender is k -distance and the adversary is $k - 1$ -distance, there are two permutations in S_{2k-1} such that, the sender can permute the identity to any of them using only k -distance but the adversary cannot reduce both to the same permutation using $k - 1$ distance.

Lemma 11. *The permutation $(k + 1, \dots, 2k - 1, k, 1, \dots, k - 1)$ and the identity permutation $(1, \dots, 2k - 1)$ cannot be both reduced to the same permutation by a $k - 1$ distance operation.*

Proof. Suppose that there exists such a permutation π . Then $\pi(1) = k$, as only k can reach the first position from both the above permutations. Similarly $\pi(2k - 1) = k$. Hence, π is no longer a permutation.

Further, in the identity permutation, $(1 \dots 2k - 1)$, only the first k elements need to be fixed. Thus for a block of size n , we can either fix the first k elements and encode the rest $n - k$ elements or apply the permutation $(k + 1, \dots, 2k - 1, k, 1, \dots, k - 1)$ and recursively encode the rest $n - 2k + 1$ elements. Thus we obtain the recurrence $P_n = P_{n-k} + P_{n-2k+1}$ for the size of the code of block size n .

The decoding strategy involves looking at the first element of the encoded permutation $p_1 = \pi(1)$. If $p_1 < k$, we can deduce that the first k elements were fixed and thus scratch out all numbers from $1 \dots k$, substitute $x - k$ for x and recursively decode the resultant string. If $p_1 > k$, we can deduce that the first $2k - 1$ elements were permuted and hence scratch them out and, substitute $x - 2k + 1$ for x and add P_{n-k} to the result of recursively decoding the resultant string.

8 Practical Results on TCP

Any communication protocol which requires packet sequence numbers can be used for steganography using our algorithms. We consider the TCP for our simulation because it is the most prevalent protocol in the Internet today. Also it is interesting to look at the interplay between TCP and our algorithms especially considering the fact that excessive packet reordering affects TCP congestion control adversely. For our purposes we use the 32-bit *Sequence Number* field in the TCP packet header. Alternatively one could also use the *Sequence Number* [5] field of the *Authentication Header* and *Encapsulating Security Payload* in the IPsec.

We performed simulations using *ns-2.28 Network Simulator* to study the behaviour of TCP under packet re-orderings. Our simulations are based on the TCP Tahoe variant. We used the BRITE topology generator for generating a 50 node 2-level hierarchical network topology which was created based on the Waxman's probability model. In this model, the probability of interconnecting two nodes u, v is given by

$$P(u, v) = \alpha e^{-d/\beta L}$$

where $0 < \alpha, \beta \leq 1$, d is the Euclidean distance from node u to v , and L is the maximum distance between any two nodes.

We chose $\alpha = 0.15, \beta = 0.2$. From the resulting topology, 25 pairs of nodes were chosen and TCP flows were started by choosing one node as a sink and the other as the source. An ftp agent was started on each of the TCP sources. Keeping this as the minimum network traffic, we performed 200 simulations choosing a pair of nodes s_i and d_i for $i \in \{1, 2, 3 \dots 200\}$, each time with s_i as the source node and d_i as the destination node. The experiment was conducted for 200 such pairs of nodes and the ratio of new throughput to the actual channel throughput (without reordering) was computed for each value of $k \in \{1, 2, 3\}$.

From the histograms thus obtained, we observe that the throughput obtained using k -distance permutations is greater than 91% for more than 68%, 60% and 30% of the source-destination pairs, for $k = 1, 2$ and 3 respectively. The corresponding average stego-information rates are 8.21bps, 11.42bps and 3.54bps. Even here, we observe that a $2 - distance$ scheme performs better than the $1 - distance$ in terms of stego-information rate, though the ratio t_r gets affected.

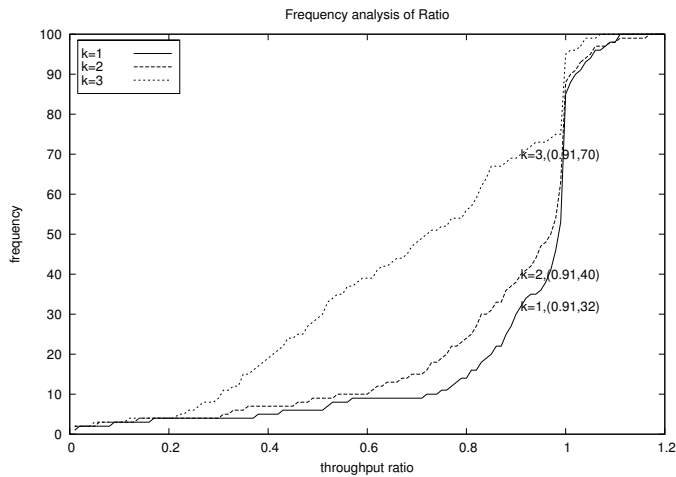


Fig. 1. Cumulative Relative Frequency of t_r

9 Conclusion

We formalize various models for packet re-ordering channels. We analyze the channel as information-theoretic game and prove the existence of Nash equilibrium. Motivated by ordered channels, eg. TCP, we introduce a new distance metric on permutations and provide error correcting codes in this metric and prove combinatorial bounds. Our codes asymptotically reach the upper bound. We simulated in detail the effects of our covert channel in various topologies and found a good correlation between the theoretical and simulated results. Being a preliminary work, this paper opens up a lot of research in this direction.

References

1. D. H. Lehmer, "Permutations with strongly restricted displacements," *Combinatorial theory and its applications II*, Eds. Erdős P., Renyi A., Sós V., pp. 755–770, 1970.
2. C. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, Illinois: University of Illinois Press, 1949.
3. I. F. Blake, "Permutation codes for discrete channels (corresp.)," *IEEE Trans. Inform. Theory*, vol. 20, pp. 138–140, Jan. 1974.
4. L. J. Schulman and D. Zuckerman, "Asymptotically good codes correcting insertions, deletions, and transpositions." *IEEE Trans. Inform. Theory*, vol. 45, no. 7, pp. 2552–2557, 1999.
5. K. Ahsan and D. Kundur, "Practical data hiding in TCP/IP," 2002. [Online]. Available: <http://citeseer.ist.psu.edu/ahsan02practical.html>
6. S. Karlin, *Mathematical Methods and Theory in Games, Programming and Economics*. Dover, TODO year, vol. 2, ch. Some chapter TODO.
7. "Steganographic communication in ordered channels," 2006. [Online]. Available: <http://abishekk.googlepages.com/stego.pdf>
8. N. S. Mendelsohn, "Permutations with confined displacements," *Canadian Math. Bulletin*, vol. 4, pp. 29–38, 1961.
9. —, "The asymptotic series for a certain class of permutation problems," *Canadian Jour. Math. B.*, vol. 8, pp. 234–244, 1956.
10. W. H. Campbell, "Indexing permutations," *J. Comput. Small Coll.*, vol. 19, no. 3, pp. 296–300, 2004.
11. G. Egorychev, "The solution of Van der Waerden's problem for permanents," *Advances in math*, vol. 42, pp. 299–305, 1981.
12. D. E. Knuth, *Fundamental Algorithms*, 2nd ed., ser. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley, 10 Jan. 1973, vol. 1, section 1.2, pp. 10–119.
13. P. Diaconis, R. Graham, and S. P. Holmes, "Statistical problems involving permutations with restricted positions," *Festschrift in Honor of William van Zwet*, 1999. [Online]. Available: <http://www-stat.stanford.edu/~susan/papers/perm8.ps>
14. M. Deza and T. Huang, "Metrics on permutations, a survey," *Journal of combinatorics, Information and System Sciences*, 1998. [Online]. Available: <http://www.liga.ens.fr/~deza/papers/voldpapers/huang/huangperm.pdf>
15. E. Zeckendorf, "Representation des nombres naturels par une somme de nombres de fibonacci ou de nombres de lucas," *Bull. Soc. Roy. Sci. Liege*, vol. 41, pp. 179–182, 1972.