A Signal Processing Perspective to Stepping-stone Detection

Ting He and Lang Tong
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853
Email: {th255,lt35}@cornell.edu

***Abstract*— Malicious use of anonymity techniques makes network attackers difficult to track. The problem is even worse in stepping-stone attacks, where multiple anonymous connections are linked to form an intrusion path. The tracking of a stepping-stone attacker requires the detection of all the connection pairs on the intrusion path. In this paper, we consider the problem of identifying a stepping-stone connection pair at an intermediate host. We formulate the problem as one of nonparametric hypotheses testing. Our attacker model allows the attacker to encrypt the traffic and modify the timing. We propose two algorithms which do not depend on the content of the traffic. Our techniques only make generic assumptions such as delay or memory constraints, and therefore they are applicable in most practical systems. We show that our algorithms can detect all the stepping-stone connections while falsely accusing normal traffic with exponentially-decaying probabilities.**

***Index Terms*— Intrusion detection, Nonparametric detection, Network security, Point processes.**

I. Introduction

Network attackers can maintain anonymity by launching the so-called stepping-stone attack [1]. In a stepping-stone attack, as illustrated in See Fig. 1, the attacker constructs a route to the victim through a collection of compromised hosts. Instead of attacking the victim directly, the attacker uses these hosts as stepping stones to relay attacking commands to the victim. Because each host can only see the identity of its immediate predecessor, the victim only sees the identity of the last host. Therefore, the identity of the attacker is hidden.

With the development of anonymity technology, attackers have more powerful ways to evade tracking. In the study of anonymity and privacy in wired networks, various techniques have been developed to help the network users to maintain anonymity while communicating efficiently. When it comes to defense against stepping-stone attacks, however, the situation becomes the other side of a coin. A malicious attacker can change the compromised hosts into his anonymity servers, and borrow ideas from existing anonymity techniques to conceal his traffic. Specifically, link level encryption can be used to change the bit pattern of the attacking packets, and other strategies can be used to change the activity of the traffic. For example, the attacker can mix his attacking traffic with other outgoing traffic at the hosts [2], impose random delay on the packets [3], change the order of packets [4], [5], or insert dummy packets [4]–[6]. Mixing attacking traffic with other traffic at the hosts puts the attacker at risk because it is easy to be detected by the destinations of the other traffic. Inserting dummy packets requires changing the content of the traffic and will not be considered in this paper.

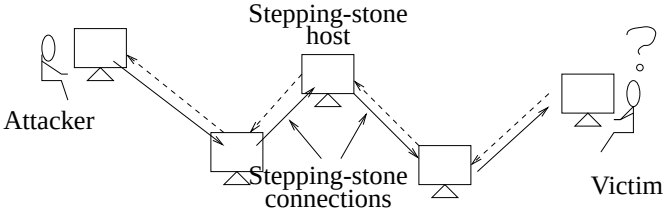

Fig. 1. A stepping-stone attack.

In this paper, we assume the attacker encrypts his traffic and pads the packets to a constant length. The attacker is allowed to change the behavior of the traffic, but not the content. Specifically, we consider packet-conserving transformations such as imposing random delay or reordering packets.

A. Related Work

Staniford and Heberlein [1] are the first to consider the problem of detecting stepping-stone connections. Early techniques are based on the content of the traffic. See, *e.g.*, [1], [7]. These techniques, however, are not applicable to detecting encrypted connections. An alternative is to exploit timing characteristics of the traffic. Zhang and Paxson [8] propose to match the "off" periods of one connection to the "on" periods of another connection. Their approach requires that the connections are synchronized. Yoda and Etoh [9] propose an algorithm to identify streams having the same traffic pattern but with possible unknown time shift. Wang, Reeves, and Wu [10] propose to correlate streams by examining packet interarrivals, and they show that their method works well if connections on different paths have distinctive timing characteristics. The

This work is supported in part by TRUST (The Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: Cisco, ESCHER, HP, IBM, Intel, Microsoft, ORNL, Qualcomm, Pirelli, Sun and Symantec, and the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. Part of this work is submitted to IEEE Transactions on Signal Processing, February 2006.

drawback of these approaches is that they are vulnerable to active timing perturbation by the attacker.

There are a few results on detecting encrypted, timing perturbed stepping-stone connections; see [11]–[13]. The key assumption of these methods is that the attacker has limited ability to alter the traffic. Donoho *et al.* [11] are the first to consider the bounded delay perturbation, where there is a maximum tolerable delay for each attacking packet. Assuming that a stepping-stone pair is a renewal process and its relay (detailed analysis is done for Poisson processes), they show that substantial correlation can be revealed even with timing perturbation. Wang and Reeves in [12] take a watermark-based approach. They show how to correlate stepping-stone connections with independent and identically distributed perturbation by introducing watermark into packet interarrivals. Blum, Song and Venkataraman [13] work along the same line as [11] except that they also assume that the attacker has a bounded peak rate, and they remove the Poisson assumption on the attacking traffic. They propose several detection algorithms with no miss detection, and they are the first to prove that their algorithms require a polynomial number of packets to satisfy certain false alarm constraint.

B. Summary of Results and Organization

We want to design a stepping-stone monitor installed either at the network gateway node (proposed by Donoho *et al.* in [11]) or as an independent process at the stepping-stone host. The monitor examines the incoming and the outgoing traffic, and identifies the stepping-stone connection pairs among normal connections. The monitor runs a hypotheses testing algorithm to test whether a pair of connections have the property of stepping-stone connections. In this paper, we consider an attacker model where the packets can be randomly delayed or permuted at the stepping-stone hosts. Two basic constraints exist for such an attacker model: the maximum packet delay and the maximum number of packets stored at a host. We derive two algorithms based on these two constraints. Our algorithms do not require synchronization of the connections, so the monitor can start working in the middle of a connection and make decisions before the connection ends. Our algorithms do not depend on either the content or the length of the packets, and therefore they are applicable to encrypted and padded connections.

Under the bounded delay constraint, we develop a timing-based algorithm which makes decisions by searching for a map between the incoming and the outgoing streams subject to the delay constraint. By restricting the search to maps that preserve the order of packets, we reduce the complexity from exponential to linear. Under the bounded memory constraint, we use a variation-based algorithm which makes decisions by comparing the maximum variation between the connections to a threshold. Both of the proposed algorithms are proved to have zero miss detection probabilities, and exponentially-decaying false alarm probabilities for independent Poisson processes. We compare the error exponents of the proposed algorithms and support our results by simulation.

The rest of the paper is organized as follows. Section II defines the problem. Section III presents two algorithms for stepping-stone detection, analyzes their performance, and compares their error exponents. Then Section IV presents simulation results to verify our analysis.

II. The Problem Statement

Let the packet arrivals on stream i be represented by a point process

$$S_i = (\ldots,\, s_{-1}^{(i)},\, s_0^{(i)},\, s_1^{(i)},\, s_2^{(i)}, \ldots), \quad i = 1,\, 2$$

where $s_k^{(i)}$ ($k \geq 1$) is the kth arrival epoch of stream i (If $k \leq 0$, it is the $(-k+1)$th packet before the monitor starts). Let $\mathcal{T}_i = \{\ldots,\, s_{-1}^{(i)},\, s_0^{(i)},\, s_1^{(i)},\, s_2^{(i)}, \ldots\}$ be the set of the elements in S_i. Let S_1 be the incoming and S_2 the outgoing streams at a particular gateway node. Normally, S_1 and S_2 are independent. If, however, S_2 is a relay of S_1 in a stepping-stone attack, then there will be strong correlation between them as formalized in the following definition.

Definition 2.1: A pair of streams $(S_1,\, S_2)$ is a *normal pair* if S_1 and S_2 are independent point processes. It is a *stepping-stone pair* if there exists a bijection $g : \mathcal{T}_1 \to \mathcal{T}_2$ such that $g(s) - s \geq 0$ for any $s \in \mathcal{T}_1$.

The bijection g is a mapping between the arrival and the departure times of packets, allowing permutation of packets during the relay[1]. The condition that g is a bijection imposes a *packet-conservation* constraint, *i.e.*, no packets are generated or dropped at the stepping stones. The condition $g(s) - s \geq 0$ is the *causality* constraint, which means that a packet cannot leave the host before it arrives.

We want to test the following binary hypotheses:

$$\begin{aligned} \mathcal{H}_0 :&\quad (S_1,\, S_2) \text{ is a normal pair,} \\ \mathcal{H}_1 :&\quad (S_1,\, S_2) \text{ is a stepping-stone pair} \end{aligned}$$

by observing $(s_1^{(i)},\, s_2^{(i)},\, s_3^{(i)}, \ldots)\,(i = 1,\, 2)$. Since no statistic property is imposed on $(S_1,\, S_2)$, the problem becomes one of nonparametric hypothesis testing. More specific assumptions will be imposed when analysis is presented.

III. Detection Algorithms and Comparison

We take an algorithmic point of view to the problem of detecting stepping-stone connections. We consider a couple of basic constraints that the attacker will encounter in concealing his attacks, and design algorithms to detect attacking traffic by testing these basic constraints.

A. Detection Based on Delay Constraint

If the attacker wants to impose random delay on packets, or change the order of packets, a critical constraint he faces is the maximum delay. In practice, long delay can cause the packets to be dropped. Furthermore, in interactive attacks, there is usually a certain order according to which the packets should arrive at the victim, and delaying the earlier packet will cause

[1]Here we do not consider the possibility of inserting chaff packets. See [11] for the description of such scenario.

all the subsequent packets to be delayed. Therefore, the delay in stepping-stone attacks is usually bounded. In this section, we consider detecting stepping-stone pairs with bounded delay, as defined in the following statement.

Definition 3.1: A pair of streams $(S_1,\ S_2)$ is a *stepping-stone pair with bounded delay* Δ if it is a stepping-stone pair, and $g(s) - s \leq \Delta$ for any $s \in \mathcal{T}_1$.

The above definition is equivalent to the definition of stepping-stone pairs proposed by Donoho *et al.* in [11].

We derive a timing-based detection algorithm "DETECT-MATCH" (DM) to detect such stepping-stone pairs. The intuition behind DM is that if we map the arrivals of a stepping-stone pair to their corresponding departures, then the mapping will satisfy causality and bounded delay. Therefore, if the detection algorithm makes decisions by searching for mappings that preserve causality and bounded delay, then we guarantee that no stepping-stone pair will be missed. For normal pairs, however, such matching may not be possible. Algorithm DM uses this property to detect stepping-stone pairs with bounded delay.

We introduce the following definitions used in the algorithm.

Definition 3.2: A *match* between $\mathcal{T}_1$ and $\mathcal{T}_2$ is a collection of pairs $\{(s_k,\ s'_k)\}_{k\in\mathbb{Z}}$ where $s_k \in \mathcal{T}_1$ and $s'_k \in \mathcal{T}_2$, such that $s_i \neq s_j$ and $s'_i \neq s'_j$ for any $i \neq j$. A length-n match $\{(s_k,\ s'_k)\}_{k=1,\ldots,\,n}$ is *valid* if $0 \leq s'_k - s_k \leq \Delta$ for all $k = 1,\ldots,\ n$. A match $\{(s_k,\ s'_k)\}_{k\in\mathbb{Z}}$ is *order-preserving* if there exists an integer m such that $s'_k = s^{(2)}_{k+m-1}$ for all k.

Since a stepping-stone pair with bounded delay must have a valid match, a direct approach is just exhaustively searching all the possible matches to find a valid one. The complexity of this approach is, however, exponential. A key observation we make here is that there may be more than one valid match for a given pair of streams, and at least one of them preserves the order of packets, as stated in the following proposition.

Proposition 3.3: If $\{(s_k,\ s'_k)\}_{k=1,\ldots,\,n}$ is a valid match, then there exists a valid match between $\{s_k\}_{k=1,\ldots,\,n}$ and $\{s'_k\}_{k=1,\ldots,\,n}$ that is order-preserving.

Proof: See Appendix. ∎

By Proposition 3.3, we see that instead of searching for any valid match, it suffices to consider only the matches that preserve the order of packets. Now the problem is reduced to finding the departure that corresponds to the first arrival, and this reduction enables us to develop a linear complexity algorithm—DM.

The detector using DM is defined as follows[2]:

$$\delta_{\text{DM}}(S_1,\ S_2,\ \Delta,\ n) = \begin{cases} 1 & \text{if } \exists m \in [h_2(s^{(1)}_1),\ h_2(\Delta)] \text{ s.t. the match} \\ & \{(s^{(1)}_k,\ s^{(2)}_{k+m-1})\}_{k=1,\ldots,\,n} \text{ is valid,} \\ 0 & \text{o.w.} \end{cases}$$

[2]The detector gives the value 1 for $\mathcal{H}_1$ and 0 for $\mathcal{H}_0$.

where $h_i(t)$ is the index of the first arrival epoch in S_i on or after time t, *i.e.*,

$$h_i(t) \stackrel{\Delta}{=} \inf\{k :\ s^{(i)}_k \geq t\}.$$

The implementation of DM is shown in Table I. The complexity of DM is at most

$$n((\#\text{ arrivals in } [s^{(1)}_1,\ \Delta) \text{ in } S_2) + 1),$$

which grows linearly with n.

TABLE I
DETECT-MATCH (DM).

```
DETECT-MATCH(S_1, S_2, Δ, n):
  for m = h_2(s_1^(1)), ..., h_2(Δ)
    for k = 1, ..., n
      if s_{k+m-1}^(2) − s_k^(1) < 0 or s_{k+m-1}^(2) − s_k^(1) > Δ   break;
    end
    if k == n + 1   return ATTACK;
  end
  return NORMAL;
```

For the performance of DM, we first show that DM has no miss detection. To see the reason, note that the match of $s^{(1)}_1$ has to satisfy certain constraints. The first constraint is causality, *i.e.*, $s^{(2)}_m \geq s^{(1)}_1$. The second constraint is that $s^{(2)}_m \leq s^{(2)}_{h_2(\Delta)}$. This is due to bounded delay condition. The first few departures can be caused by arrivals before $s^{(1)}_1$, but departures after Δ have to correspond to arrivals on or after $s^{(1)}_1$. Hence, m has to satisfy $h_2(s^{(1)}_1) \leq m \leq h_2(\Delta)$, as shown in Fig. 2. This condition combined with Proposition 3.3 guarantees that all stepping-stone pairs with bounded delay Δ will be detected by δ_{DM}.

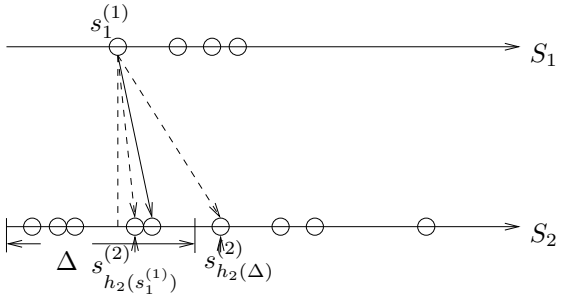

Fig. 2. The match of $s^{(1)}_1$: there are three possible candidates.

Next, we show that if normal pairs are pairs of independent Poisson processes, the false alarm probability of DM goes to zero exponentially, as stated in the following theorem.

Theorem 3.4: If normal pairs are independent Poisson processes of equal rate λ, then the false alarm probability of DM is bounded by

$$P_F(\delta_{\text{DM}}) \leq \min\left\{\frac{3}{2} - e^{-\lambda\Delta}(\frac{3}{2} + \lambda\Delta),\ 1\right\}\gamma^{n-1}$$

where $\gamma = 1 - e^{-\lambda\Delta/2}$.

Proof: See Appendix. ∎

Remark: Note that $\gamma \to 0$ as $\lambda \to 0$. Therefore, the false alarm probability of DM decays much faster for slow traffic. An intuitive explanation is that $\lambda \to 0$ means the inter-arrival time $\to \infty$, which, when the ratio between inter-arrival time and delay is considered, is equivalent to having finite inter-arrival time but $\Delta \to 0$. That is, for extremely slow traffic, DM raises alarms only if the departure times match the arrival times perfectly, and this is unlikely to happen between two independent connections. On the other hand, if $\lambda \to \infty$, then Δ is far larger than inter-arrival times, *i.e.,* the delay constraint is essentially removed. Therefore, DM will always raise alarms in that case.

B. Detection Based on Memory Constraint

Another basic constraint for the attacker is how much memory he can use on the host. No matter whether he chooses to randomize the departure times or the departure order, there is one fundamental constraint that he can not hide—the memory usage. In the random delay strategy, packets have to be stored in the host while waiting for the delay to expire. In the permutation strategy, packets have to be put in a pool while the host is waiting for more packets to mix in a batch. We notice that by looking at the number of arrivals and departures at the two ends of a host, a monitor can still discover consistent property about the stepping-stone pairs without knowing what happens to the packets in the host. Specifically, assume that the host's memory can hold at most M packets. Then the difference between the number of incoming and the number of outgoing packets during any period of time can never exceed M. We abstract this property to the following definition.

Definition 3.5: A pair of streams $(S_1,\ S_2)$ is a *stepping-stone pair with bounded memory* M if it is a stepping-stone pair, and for any $a \le b$,

$$\big|\big|\{s \in \mathcal{T}_1 :\ s \in [a,\ b]\}| - |\{s \in \mathcal{T}_2 :\ s \in [a,\ b]\}|\big| \le M.$$

In [14], we have proposed an algorithm, "DETECT-MAXIMUM-VARIATION" (DMV), to detect stepping-stone pairs by comparing the maximum variation with a predefined threshold[3]. For stepping-stone pairs with bounded memory M, the detector is defined as

$$\delta_{\text{DMV}}(S_1,\ S_2,\ M,\ n) = \begin{cases} 1 & \text{if } v(n) \le M, \\ 0 & \text{o.w.} \end{cases}$$

where $v(n)$ is the maximum variation computed by DMV.

We will show in the sequel why DMV can be applied to stepping-stone with bounded memory and its performance. We first show that any stepping-stone pairs with bounded memory M will be detected by DMV. The reason follows from a reinterpretation of the maximum variation. In [14], the maximum variation $v(w)$ is defined as follows

$$v(w) \stackrel{\Delta}{=} \max_{1 \le i \le w} d(i) - \min_{1 \le i \le w} d(i),$$

where $d(w) \stackrel{\Delta}{=} N_1(w) - N_2(w)$, and $N_i(w) \stackrel{\Delta}{=} \sum_{j=1}^{w} I_{\{s_j \in S_i\}}$. Note that $(s_1,\ s_2,\ s_3, \ldots)$ is the ordered union of S_1 and S_2. By simple manipulation, we see that

$$v(w) = \max_{1 \le i \le j \le w} |(N_1(j) - N_1(i)) - (N_2(j) - N_2(i))|.$$

Therefore, $v(n)$ is the maximum difference in the number of arrivals and the number of departures in any period of time up to the nth packet (either an arrival or a departure). By Definition 3.5, $v(n)$ is bounded by M.

As for the false alarm probability of DMV, we refer to the following theorem in [14]:

Theorem 3.6: If normal pairs consist of independent Poisson processes, then the false alarm probability of DMV is bounded by

$$P_F(\delta_{\text{DMV}}) \le \frac{(M+1)}{1-\rho}\rho^n,$$

where $\rho = \cos\frac{\pi}{M+2}$. Furthermore, the upper bound is tight with respect to the error exponent, *i.e.,*

$$\lim_{n\to\infty} -\frac{1}{n}\log P_F(\delta_{\text{DMV}}) = -\log\rho.$$

C. Comparing the Algorithms

In practice, the attacker's strategy may cause his traffic to satisfy both the bounded delay and the bounded memory conditions. We are interested in which algorithm performs better in detecting such stepping-stone pairs[4].

Theorems 3.6 and 3.4 suggest that DM is preferable if $\gamma \le \rho^2$, *i.e.,* when the rate λ of the normal traffic satisfies

$$\lambda \le -\frac{4}{\Delta}\log\left(\sin\frac{\pi}{M+2}\right). \tag{1}$$

Otherwise, DMV is preferable.

The threshold rate estimated in (1) may not be exact because Theorem 3.4 only gives a lower bound on the error exponent of δ_{DM}. We believe, however, that the existence of such a threshold rate is true in general because on the one hand, rate does not affect the performance of DMV, and on the other, the performance of DM changes with rate as argued in Subsection III-A.

IV. Simulations

We implement DM and DMV to verify their performance. Since each algorithm has no miss detection under its own assumptions, we are only interested in false alarm probabilities. In our simulation, we assume that normal pairs are independent Poisson processes of equal rates. The exact rate will be specified later.

We compare DMV and DM by plotting their simulated false alarm probabilities together in Fig. 3. We set $M = 4$, $\Delta = 1$, and simulate DM for several traffic rates $\lambda = 2,\ 3,\ 4,\ 5$ (note that $P_F(\delta_{\text{DMV}})$ does not depend on rate). Our simulation verifies that $P_F(\delta_{\text{DM}})$ increases with the growth of traffic rate, as argued

[3]In [14], it is proposed to use DMV to detect stepping-stone pairs with bounded delay and bounded number of arrivals within the maximum delay, which is a subset of stepping-stone pairs with bounded memory

[4]Note that for large n, $\delta_{\text{DM}}(S_1,\ S_2,\ \Delta,\ n)$ uses approximately twice as many samples as $\delta_{\text{DMV}}(S_1,\ S_2,\ M,\ n)$. Thus we should compare $\delta_{\text{DMV}}(S_1,\ S_2,\ M,\ 2n)$ with $\delta_{\text{DM}}(S_1,\ S_2,\ \Delta,\ n)$.

in Subsection III-A. The plot clearly shows a threshold on the rate beyond which DMV outperforms DM. In our simulation, the threshold rate estimated by (1) is about 2.7726, whereas the actual threshold rate is shown by the simulation to be somewhere between 3 and 4. Therefore, we conclude that the lower bound on the error exponent of δ_{DM} obtained from Theorem 3.4 is not tight. We expect the actual threshold rate to be larger than the one estimated by (1).

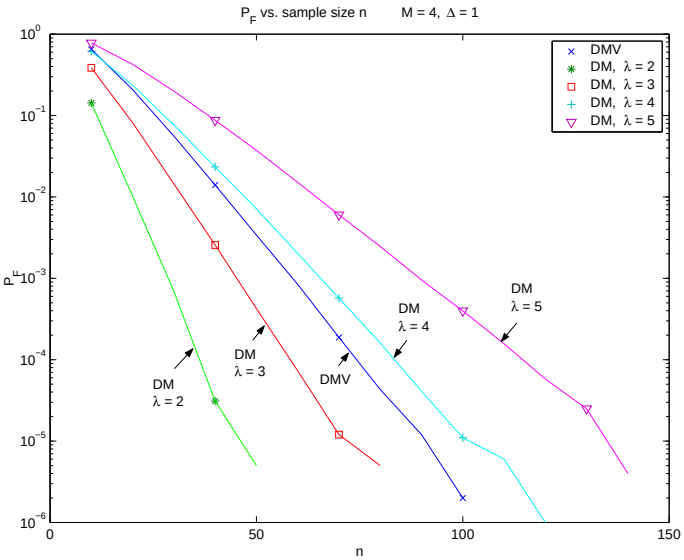

Fig. 3. $P_F(\delta_{\text{DMV}})$ and $P_F(\delta_{\text{DM}})$.

APPENDIX

Proof of Proposition 3.3

As illustrated in Fig. 4, if $\{(s_1,\, s_1'),\, (s_2,\, s_2')\}$ is a valid match which does not preserve the order of packets, we can switch the match to be $\{(s_1,\, s_2'),\, (s_2,\, s_1')\}$ such that it is still valid but the order is preserved. By this idea, we can reorder the match as

$$s_k'' = \min_{k\le i\le n} s_i'.$$

Then $\{(s_k,\, s_k'')\}_{k=1,\ldots,\,n}$ is valid and order-preserving.

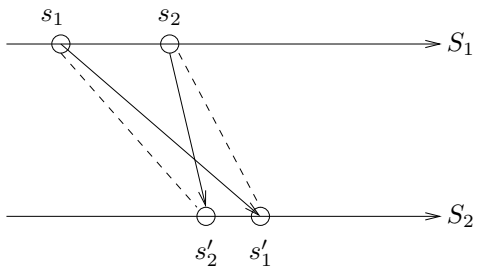

Fig. 4. More than one valid match: both the solid and the dotted lines are valid matches.

■

Proof of Theorem 3.4

Given a match $\{(s_i,\, s_i')\}_{i=1,\,2,\ldots}$, define $Y_i \triangleq s_i' - s_i$. Algorithm DM has a false alarm if and only if there exists s_1' *s.t.* the order-preserving match $\{(s_i, s_i')\}_{i=1,\ldots,\,n}$ satisfies $0 \le Y_i \le \Delta$ for all $i = 1, \ldots,\, n$.

Define Z_i $(i \ge 2)$ as the difference in the ith interarrival time, *i.e.*,

$$Z_i \triangleq (s_i' - s_{i-1}') - (s_i - s_{i-1}).$$

Then

$$Y_i = (s_{i-1}' - s_{i-1}) + (s_i' - s_{i-1}') - (s_i - s_{i-1}) = Y_{i-1} + Z_i.$$

Note that for independent Poisson processes of equal rate λ, $s_i' - s_{i-1}'$ and $s_i - s_{i-1}$ are independent Exponential random variables with mean $1/\lambda$. Thus Z_i's $(i \ge 2)$ are i.i.d. random variables with p.d.f. $p(z) = \frac{\lambda}{2} e^{-\lambda|z|}$. Therefore, given $Y_1 = y_1$, $\{Y_i\}_{i=2}^{\infty}$ is a general random walk starting from y_1 with step distribution $p(z)$.

Let $Y_2^n \triangleq (Y_2, \ldots,\, Y_n)$. Since Z_i has zero mean, it is easy to see that

$$\begin{aligned}\Pr\{Y_2^n \in [0,\, \Delta] | Y_1 \in [0,\, \Delta]\} &\le \Pr\{Y_2^n \in [0,\, \Delta] | Y_1 = \frac{\Delta}{2}\} \\ &= \Pr\{|Y_2^n| \le \frac{\Delta}{2} | Y_1 = 0\}.\end{aligned}$$

We see that the false alarm probability satisfies

$$\begin{aligned}P_F(\delta_{\text{DM}}) &= \Pr\{\exists s_1',\ \textit{s.t.}\ 0 \le Y_1^n \le \Delta\} \\ &\le \Pr\{\exists s_1',\ \textit{s.t.}\ 0 \le Y_1 \le \Delta\} \\ &\quad \cdot \Pr\{|Y_2^n| \le \frac{\Delta}{2} | Y_1 = 0\}. \qquad (2)\end{aligned}$$

We first bound the second term in (2). Define

$$p_n(z)dz \triangleq \Pr\{|Y_2^{n-1}| \le \frac{\Delta}{2},\ z < Y_n < z + dz | Y_1 = 0\},$$

where $n = 2,\, 3, \ldots$ and $-\infty < z < \infty$. Note that

$$\Pr\{|Y_2^n| \le \frac{\Delta}{2} | Y_1 = 0\} = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} p_n(z)dz.$$

In [15] it is shown that

$$p_n(z) = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} p_{n-1}(x) p(z-x) dx \ \ (n = 2,\, 3, \ldots)$$

where $p_1(z) = \delta(z)$ (Dirac delta function), and $p(z)$ is the step distribution. Then we have

$$\begin{aligned}&\Pr\{|Y_2^n| \le \frac{\Delta}{2} | Y_1 = 0\} \\ =\ & \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} p_n(z_n) dz_n \\ =\ & \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} p_{n-1}(z_{n-1}) dz_{n-1} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} p(z_n - z_{n-1}) dz_n \\ =\ & \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} p(z_2) dz_2 \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} p(z_3 - z_2) dz_3 \cdots \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} p(z_n - z_{n-1}) dz_n\end{aligned}$$

Let $\gamma \stackrel{\Delta}{=} \max_{c \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]} \int_{-\frac{\Delta}{2}-c}^{\frac{\Delta}{2}-c} p(z)dz$. Simple calculation yields that $\gamma = 1 - e^{-\lambda\Delta/2}$. Then

$$\Pr\{|Y_2^n| \leq \frac{\Delta}{2} | Y_1 = 0\} \leq \gamma^{n-1}.$$

Now we bound the first term in (2) as follows:

$$\begin{aligned}\Pr\{\exists s_1', \textit{ s.t. } 0 \leq Y_1 \leq \Delta\} \quad \leq \quad & \Pr\{\exists \text{ departures } \in [s_1,\, \Delta)\} \\ & + \Pr\{0 \leq s_{h_2(\Delta)}^{(2)} - s_1 \leq \Delta\}.\end{aligned}$$

This is a union bound: the first probability corresponds to the case when s_1' is some packet in S_2 between s_1 and Δ, and the second corresponds to the case when $s_1' = s_{h_2(\Delta)}^{(2)}$. It is easy to show that

$$\begin{aligned}&\Pr\{\exists \text{ departures } \in [s_1,\, \Delta)\} \\ &\qquad = \int_0^{\Delta} \lambda e^{-\lambda s_1}(1 - e^{-\lambda(\Delta - s_1)})ds_1 \\ &\qquad = 1 - (1 + \lambda\Delta)e^{-\lambda\Delta}.\end{aligned}$$

Since $s_{h_2(\Delta)}^{(2)} - s_1 \stackrel{d}{=} \Delta + Z_i$,

$$\Pr\{0 \leq s_{h_2(\Delta)}^{(2)} - s_1 \leq \Delta\} = \int_{-\Delta}^{0} p(z)dz = \frac{1}{2}(1 - e^{-\lambda\Delta}).$$

Hence

$$\Pr\{\exists s_1', \textit{s.t. } 0 \leq Y_1 \leq \Delta\} \leq \min\left\{\frac{3}{2} - e^{-\lambda\Delta}(\frac{3}{2} + \lambda\Delta),\, 1\right\}.$$

Therefore, by (2) we have proved that

$$P_F(\delta_{\text{DM}}) \leq \min\left\{\frac{3}{2} - e^{-\lambda\Delta}(\frac{3}{2} + \lambda\Delta),\, 1\right\}\gamma^{n-1}$$

where $\gamma = 1 - e^{-\lambda\Delta/2}$ satisfying $0 < \gamma < 1$.

■

REFERENCES

[1] S. Staniford-Chen and L. Heberlein, "Holding intruders accountable on the internet," in *Proc. the 1995 IEEE Symposium on Security and Privacy*, (Oakland, CA), pp. 39–49, May 1995.

[2] D. Chaum, "Untraceable electronic mail, return addresses and digital pseudonyms," *Communications of the ACM*, vol. 24, pp. 84–88, February 1981.

[3] D. Kesdogan, J. Egner, and R. Buschkes, "Stop-and-go MIXes providing probabilistic security in an open system," in *Second International Workshop on Information Hiding (IH'98), Lecture Notes in Computer Science*, vol. 1525, (Portland, Oregon), pp. 83–98, April 1998.

[4] O. Berthold, H. Federrath, and S. Kopsell, "Web MIXes: A system for anonymous and unobservable Internet access," in *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability, Lecture Notes in Computer Science*, vol. 2009, (Berkeley, CA), pp. 115–129, July 2000.

[5] A. Pfitzmann, B. Pfitzmann, and M. Waidner, "ISDN-MIXes: Untraceable communication with very small bandwidth overhead," in *Proceedings of the GI/ITG Conference: Communication in Distributed Systems, Informatik-Fachberichte*, vol. 267, (Mannheim, Germany), pp. 451–463, February 1991.

[6] O. Berthold and H. Langos, "Dummy traffic against long term intersection attacks," in *Proceedings of the Privacy Enhancing Technologies Workshop (PET'02), Lecture Notes in Computer Science*, vol. 2482, (San Francisco, CA), pp. 110–128, April 2002.

[7] X. Wang, D. Reeves, S. Wu, and J. Yuill, "Sleepy watermark tracing: An active network-based intrusion response framework," in *Proc. of the 16th International Information Security Conference*, pp. 369–384, 2001.

[8] Y. Zhang and V. Paxson, "Detecting stepping stones," in *Proc. the 9th USENIX Security Symposium*, pp. 171–184, August 2000.

[9] K. Yoda and H. Etoh, "Finding a connection chain for tracing intruders," in *6th European Symposium on Research in Computer Security, Lecture Notes in Computer Science 1895*, (Toulouse, France), October 2000.

[10] X. Wang, D. Reeves, and S. Wu, "Inter-packet delay-based correlation for tracing encrypted connections through stepping stones," in *7th European Symposium on Research in Computer Security, Lecture Notes in Computer Science 2502*, pp. 244–263, 2002.

[11] D. Donoho, A. Flesia, U. Shankar, V. Paxson, J. Coit, and S. Staniford, "Multiscale stepping-stone detection: Detecting pairs of jittered interactive streams by exploiting maximum tolerable delay," in *5th International Symposium on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science 2516*, 2002.

[12] X. Wang and D. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of inter-packet delays," in *Proc. of the 2003 ACM Conference on Computer and Communications Security*, pp. 20–29, 2003.

[13] A. Blum, D. Song, and S. Venkataraman, "Detection of Interactive Stepping Stones: Algorithms and Confidence Bounds," in *Conference of Recent Advance in Intrusion Detection (RAID)*, (Sophia Antipolis, French Riviera, France), September 2004.

[14] T. He and L. Tong, "Detecting Encrypted Interactive Stepping-stone Connections," in *Proc. 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Toulouse, France), May 2006.

[15] D. Cox and H. Miller, *The Theory of Stochastic Processes*. New York: John Wiley & Sons Inc., 1965.