Detecting Encrypted Stepping-stone Connections

Ting He, *Student Member, IEEE,* and Lang Tong†, *Fellow, IEEE*

***Abstract*— Stepping-stone attacks are often used by network intruders to hide their identities. In a stepping-stone attack, attacking commands are sent indirectly to the victim through a chain of compromised hosts acting as "stepping stones". In defending against such attacks, it is necessary to detect stepping-stone connections at the compromised hosts. The use of encrypted connections by the attacker complicates the detection problem, and the attacker's active timing perturbation and insertion of chaff make it even more challenging. This paper considers strategies to identify stepping-stone connections when the attacker is able to encrypt the attacking packets and perturb their timing. Furthermore, the attacker can also add chaff packets in the attacking stream. The paper first considers stepping-stone connections subject to packet-conserving transformations by the attacker. Two activity-based algorithms are proposed to detect stepping-stone connections with bounded memory or bounded delay perturbation, respectively. These algorithms are proved to have exponentially decaying false alarm probabilities if normal traffic can be modelled as Poisson processes. It is shown that the proposed algorithms improve the performance of an existing stepping-stone detection algorithm. The paper then addresses the detection of stepping-stone connections with both timing perturbation and chaff. Robust algorithms are developed to deal with chaff evasion. It is proved that the proposed robust algorithms can tolerate a number of chaff packets proportional to the size of the attacking traffic, and have vanishing false alarm probabilities for Poisson traffic. Simulations using synthetic data are used to validate the theoretical analysis. Further results using actual Internet traces are shown to demonstrate the performance of the proposed algorithms.**

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

I. Introduction

To evade surveillance, network attackers can hide their identity by launching the so-called stepping-stone attack [1]. In such an attack, as illustrated in See Fig. 1, the attacker compromises a collection of hosts and uses these hosts as stepping stones to relay attacking commands. Because each connection is made by a separate remote login, a host in the chain can only see the identity of its immediate predecessor, and the victim only sees the identity of the last host. Therefore, the identification of attackers requires tracing the chain of stepping stones. A key component in such tracing is the detection of stepping stone connections.

†Corresponding author.

T. He and L. Tong are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853. Email: {th255@, ltong@ece.}cornell.edu.

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 presented in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2006, and Conference on Information Sciences and Systems (CISS) 2006.

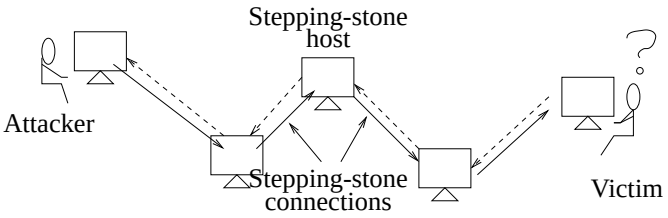

Fig. 1. A stepping-stone attack.

Donoho *et al.* proposed in [2] the use of stepping-stone monitors at each gateway node for detection. A pair of incoming and outgoing streams is called a *stepping-stone pair* if it is part of a stepping-stone attack. Otherwise, it is called a *normal pair*[1]. The stepping-stone monitors try to discover all stepping-stone pairs by examining the incoming-outgoing traffic. In practice, the monitor has to make decisions by observing live traffic, which may not include the beginning or the end of the connection. Therefore, it is desirable that the detection strategy does not require synchronization between incoming and outgoing streams. Besides, the connections may be encrypted (*e.g.,* SSH sessions) so that the monitor cannot rely on the content of the traffic. Furthermore, a careful attacker may even actively modify the traffic each time it passes through a host in order to confuse the monitor.

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], [3]. These techniques, however, are not applicable to detecting encrypted connections. An alternative is to exploit timing characteristics of the traffic. Zhang and Paxson [4] propose to detect stepping-stone connections by matching the ending of "off" periods in different connections. Their approach requires that the connections are synchronized. Yoda and Etoh [5] propose an algorithm to identify streams with the same traffic pattern but unknown time shift. Wang, Reeves, and Wu [6] propose to correlate streams by examining packet interarrival times, and they show that their method works well if connections on different paths have distinctive timing characteristics. The 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 [2], [7], [8]. The key assumption of these methods is that the attacker is able to perform a packet-conserving transformation on his traffic, but the transformation is subject to certain constraints.

[1]Formal definitions are given in Section II.

Donoho *et al.* [2] 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 [7] take a watermark-based approach. They show how to correlate stepping-stone connections with independent and identically distributed perturbation by introducing watermark into packet interarrival times. Blum *et al.* [8] work along the same line as [2] 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 a detection algorithm called "DETECT-ATTACKS" (DA) with no miss, and they are the first to prove that their algorithm requires a polynomial number of packets to satisfy certain false alarm constraint.

A more general category of stepping-stone connections is the one allowing non-packet-conserving transformations. Here the attacker has the ability to mix attacking traffic with non-attacking traffic, including dummy traffic called chaff, to evade detection, or he can repacketize his traffic so that there is no 1-1 correspondence between arriving and departing packets. The repacketization is outside the scope of packet level detection, and should be addressed at a lower level; the insertion of chaff, however, has to be dealt with effectively. Peng *et al.* in [9] propose an active detection scheme which combines watermarking with packet matching to detect stepping-stone traffic in chaff. They assume packets have bounded delays, and chaff only appears in the downstream flow. Their scheme injects watermarks in the upstream flow, and finds a subsequence in the downstream flow, whose watermark is closest to the injected one. Such an active scheme, however, requires the control of the stepping-stone host, and it also reveals the activities of the detector to the attacker, allowing the attacker to compromise the detector by studying its behavior. Donoho *et al.* [2] point out that in principle it is possible to correlate stepping-stone traffic even if both (bounded) delay and independent chaff are introduced during the relay. Blum *et al.* [8] modify their algorithm DA into a new algorithm called "DETECT-ATTACKS-CHAFF" (DAC) to deal with chaff. DAC detects stepping-stone traffic with a limited number of chaff packets by increasing the detection threshold. The drawback is that such an increase in the threshold leads to an increase in the false alarm probability, and the attacker can still evade detection by adding an arbitrarily small fraction of chaff traffic. Indeed, a fixed number of chaff packets can evade the detection for an attacking traffic of arbitrary size. In a recent paper [10], Zhang *et al.* propose packet matching schemes to detect stepping-stone traffic with bounded delay perturbation and/or chaff. For a stepping-stone traffic with bounded delay but without chaff, they propose a detection strategy similar to "DETECT-MATCH" [11], although the detection performance on attacking traffic is not proved, and they do not have a closed form characterization for the false alarm probability. For stepping-stone traffic with both bounded delay and chaff, they propose a matching strategy which can detect stepping-stone traffic if the chaff is only inserted in the departing stream. They prove that this strategy has exponentially decaying false alarm probability for independent Poisson streams.

B. Summary of Results and Organization

In this paper, we consider the problem of detecting encrypted stepping-stone connections subject to the attacker's active modification. Our strategy does not use the content of the traffic. Nor is synchronization or active traffic manipulation required. We first consider detecting stepping-stone pairs with bounded perturbation but no chaff, and then generalize our detection schemes to handle chaff packets. We formulate the problem of detecting stepping-stone connections as a hypothesis testing of independent against correlated point processes. For the traffic perturbation by the attacker, we consider two types of constraints: (i) the host has bounded memory; (ii) attacking packets have bounded delay. While the bounded delay condition is a key in [2], [8]–[10], the bounded memory constraint, to the best of our knowledge, has not been addressed in the literature.

Under the bounded memory assumption, we develop a linear complexity algorithm based on the maximum variation statistic. The intuition behind this algorithm is that the maximum variation statistic stays bounded for relayed traffic going through a stepping-stone host with limited memory, but diverges unboundedly for independent traffic. Under the bounded delay assumption, we derive a timing-based algorithm based on the idea of matching arriving packets with departing packets. By restricting the search to maps that preserve the order of packets, we reduce the complexity from exponential to linear. We prove that both of the proposed algorithms have no miss for their targeting stepping-stone pairs, and exponentially decaying false alarm probabilities for independent Poisson processes. We then generalize the attacker model to allow the presence of chaff. We develop two new algorithms for stepping-stone pairs with both bounded memory or bounded delay perturbation and chaff. The idea is to declare a stream pair normal if the optimal chaff-inserting algorithm would have had to insert a certain fraction of chaff packets to embed attacking packets into the given stream pair. Therefore, the attacker will have to insert at least the same fraction of chaff to evade detection. The threshold on the fraction of chaff is chosen to be as large as possible to make the attacker's evasion difficult, but also small enough so that the false alarm probability will go to zero as the traffic size increases.

We next compare the performance of existing algorithms and the proposed algorithms. To make the comparison, we analyze the performance of algorithms DA and DAC proposed by Blum *et al.* [8]. The original analysis by Blum *et al.* focuses on sample size, whereas our result is on error exponent analysis. Among algorithms dealing with packet-conserving transformations, we show that the proposed variation-based algorithm has larger false alarm error exponent than the algorithm DA by Blum *et al.* , and it also outperforms the proposed matching-based algorithm when the traffic is sufficiently fast. For slow traffic, however, the matching-based algorithm can be much better. For algorithms dealing with chaff, we compare

our algorithms with existing algorithms by Blum *et al.* and Zhang *et al.* [8], [10]. We show that, in contrast to the constant chaff tolerance of the algorithms by Blum *et al.* and Zhang *et al.* , our algorithms are capable of handling an amount of chaff growing linearly with the size of the attacking traffic. The price we paid is that the false alarm probabilities of our algorithms no longer have exponentially decaying upper bounds, but are only guaranteed to be vanishing asymptotically.

The rest of the paper is organized as follows. Section II defines the detection problem. Section III presents a variation-based algorithm and its performance analysis under the bounded memory assumption. Section IV develops a timing-based algorithm for stepping-stone pairs with bounded delay and analyzes its performance. In Section V, we present robust algorithms for stepping-stone pairs with both bounded perturbation and chaff, and analyze their robustness and asymptotic false alarm probabilities. Section VI compares the performance of the proposed algorithms with several existing algorithms for detecting stepping-stone pairs with or without chaff, respectively. Section VII gives simulation results on both synthetic data and internet traces to verify the performance. The paper is concluded by Section VIII with a few remarks on the application of such detection schemes.

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. 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.

If only a subsequence of S_i ($i = 1,\, 2$) consists of attacking packets, then only that part is constrained, as stated in the following definition.

Definition 2.2: A pair of streams $(S_1,\, S_2)$ is a *stepping-stone pair with chaff* if it is the superposition of a stepping-stone pair $(S_1',\, S_2')$ and a pair of chaff streams $(C_1,\, C_2)$ (either or both of them can be empty).

Stream C_i ($i = 1,\, 2$) consists of dummy packets called *chaff* which do not need to arrive at the victim. Chaff packets can be generated or dropped at any stepping stone hosts without affecting the attack. They are artificially inserted by the attacker to evade detection.

We want to test the following binary hypotheses:

$\mathcal{H}_0:\ (S_1,\, S_2)$ is a normal pair,

$\mathcal{H}_1:\ (S_1,\, S_2)$ is a stepping-stone pair (with or without chaff),

by observing $(s_1^{(i)},\, s_2^{(i)},\, s_3^{(i)}, \ldots)\,(i = 1,\, 2)$. This is a nonparametric hypothesis testing problem; no specific assumptions on the statistical properties of $(S_1,\, S_2)$ are imposed at this point. Additional assumptions on normal and stepping-stone pairs will be introduced later when detailed detection algorithms and analysis are presented.

III. Detecting Stepping-stone Pairs with Bounded Memory

We consider the problem of detecting stepping-stone pairs when the host has bounded memory. Specifically, assume that the host's memory can hold at most M packets[2]. Then the difference between the number of incoming and the number of outgoing packets during any period can never exceed M. We use this property to define such stepping-stone pairs as follows.

Definition 3.1: 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 $a \leq b$,

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

To detect stepping-stone pairs with bounded memory, we derive a counting-based algorithm—DETECT-MAXIMUM-VARIATION (DMV).

Before presenting the algorithm, we need to introduce some definitions. Merge $(s_1^{(1)},\, s_2^{(1)}, \ldots)$ and $(s_1^{(2)},\, s_2^{(2)}, \ldots)$ and order the union as $(s_1,\, s_2,\, s_3, \ldots)$. Let $N_i(w)$ $(i = 1,\, 2)$ be the number of packets monitored in S_i when the total number of monitored packets is w, *i.e.,*

$$N_i(w) \triangleq \sum_{j=1}^{w} I_{\{s_j \in S_i\}},$$

where $I_{\{\cdot\}}$ is the indicator function. Sample paths of $N_1(w)$ and $N_2(w)$ are illustrated in Fig. 2. (a).

Define the cumulative difference between S_1 and S_2 as

$$d(w) \triangleq N_1(w) - N_2(w),$$

and let the maximum variation of $d(w)$ be

$$v(w) \triangleq \max_{1 \leq i \leq w} d(i) - \min_{1 \leq i \leq w} d(i).$$

See Fig. 2. (b) for an illustration of $d(w)$ and $v(w)$.

If the stepping-stone host has bounded memory, then the sample path of $d(w)$ will have bounded variation. Algorithm DMV distinguishes normal and stepping-stone pairs by looking at the maximum variation. Specifically, note that

$$\begin{aligned} v(w) &= \max_{1 \leq i \leq j \leq w} |d(j) - d(i)| \\ &= \max_{1 \leq i \leq j \leq w} |(N_1(j) - N_1(i)) - (N_2(j) - N_2(i))|, \end{aligned}$$

[2]Similar requirement on buffer size has been considered by Giles and Hajek in the context of timing channels [12].

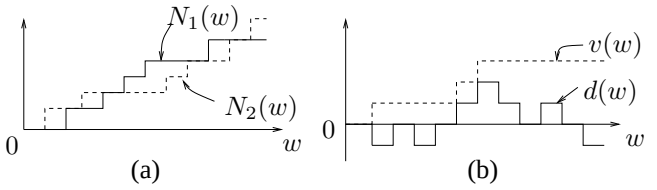

Fig. 2. (a) the cumulative counts $N_i(w)$ $(i = 1, 2)$; (b) the cumulative difference $d(w)$ and the maximum variation $v(w)$.

where $|(N_1(j)-N_1(i))-(N_2(j)-N_2(i))|$ is the difference in the number of incoming and the number of outgoing packets between the ith and the jth arrivals. For stepping-stone pairs with bounded memory M, this difference is bounded by M, *i.e.,* $v(w) \le M$, $\forall w$. Using the maximum variation statistics, we define the following detector[3]:

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

The detector can be implemented by the algorithm DMV shown in Table I.

TABLE I
DETECT-MAXIMUM-VARIATION (DMV).

DETECT-MAXIMUM-VARIATION($S_1,\, S_2,\, M,\, n$):
  $d_{\max} = d_{\min} = 0$;
  for $w = 1 : n$
    $d(w) = \begin{cases} d(w-1)+1 & \text{if } s_w \in \mathcal{T}_1 \\ d(w-1)-1 & \text{if } s_w \in \mathcal{T}_2 \end{cases}$;
    $d_{\max} = \max(d_{\max},\, d(w))$;
    $d_{\min} = \min(d_{\min},\, d(w))$;
    if $d_{\max} - d_{\min} > M$ return NORMAL;
  end
  return ATTACK;

Algorithm DMV has complexity $O(n)$ and uses only constant memory ($O(\log M)$, to be precise[4]). Any stepping-stone pair with bounded memory M will be detected after n packets, *i.e.,* miss is totally avoided. We only need to be concerned about the false alarm probability, and it is bounded as follows.

Theorem 3.2: 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, if the two Poisson processes have the same rates, then the upper bound is tight with respect to the error exponent, *i.e.,*

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

Proof: See Appendix. ∎

Remarks: For a given false alarm probability α, we can guarantee that the false alarm probability of DMV is bounded by α by making its upper bound equal to α, yielding a sample size

$$n = \frac{\log\alpha(1-\rho) - \log(M+1)}{\log\rho} \tag{1}$$

which grows as $O(M^2 \log\frac{M}{\alpha})$ as $M \to \infty$ and $\alpha \to 0$. For example, if $M = 20$, (1) says that using 1196 packets will guarantee a false alarm probability no greater than 1%.

IV. Detecting Stepping-stone Pairs with Bounded Delay

Many stepping-stone attacks are interactive. In interactive stepping-stone attacks, the attacker waits for the feedback of the previous commands and sends new commands based on the feedback. Therefore, the delay in such interactive attacks is usually bounded. In this section, we consider detecting stepping-stone pairs with bounded delay, which is defined as follows.

Definition 4.1: 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 $g(s) - s \le \Delta$ for any $s \in \mathcal{T}_1$.

Our definition of stepping-stone pair with bounded delay is the same as the one proposed by Donoho *et al.* in [2]. The bounded delay model is fundamentally different from the bounded memory model considered in Section III. It has been shown that the two models have very different scaling behavior on the mutual information between the incoming stream and the outgoing stream [12]. In Section VI-A.2, we will show that they also have difference detection performance with respect to changes in traffic rates.

We derive a timing-based detection algorithm DETECT-MATCH (DM) to detect such stepping-stone pairs. Algorithm DM matches the first n packets in S_1 with their possible relays in S_2, subject to the maximum delay Δ. For stepping-stone pairs with bounded delay, there must be at least one way of matching that satisfies causality and bounded delay constraints—matching the arrivals of packets with the departures of the same packets. For normal pairs, however, such matching may not be possible. Algorithm DM uses this property to detect stepping-stone pairs with bounded delay.

A few definitions are needed to present the algorithm. Define $h_i(t)$ to be the index of the first arrival epoch in S_i on or after time t, *i.e.,*

$$h_i(t) \stackrel{\Delta}{=} \inf\{k :\, s_k^{(i)} \ge t\}.$$

For example, $s^{(2)}_{h_2(\Delta)}$ is the first epoch in S_2 on or after time Δ (see Fig. 4).

Definition 4.2: 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 $s_i \ne s_j$ and $s'_i \ne s'_j$ for any $i \ne j$. A length-n match $\{(s_k, s'_k)\}_{k=1}^n$ is *valid* if $0 \le s'_k - s_k \le \Delta$ for all $k = 1, \ldots, n$. A match $\{(s_k,\, s'_k)\}_{k\in\mathbb{Z}}$ is *order-preserving* if $s_k \le s_l$ implies $s'_k \le s'_l$ for all $k,\, l$.

From this definition, it is easy to see that a stepping-stone pair with bounded delay must have at least one valid match. Thus one way to detect such stepping-stone pairs is by looking for a valid match between the arrivals and the departures. The

[3]We use the convention that the detector gives the value 1 for $\mathcal{H}_1$ and 0 for $\mathcal{H}_0$.

[4]The log in this paper is always natural logarithm.

complexity of this approach is, however, exponential[5]. Instead of searching for any valid match, we prove that it suffices to limit our search to order-preserving, valid matches, as stated in the following proposition.

Proposition 4.3: If $\{(s_k,\ s'_k)\}_{k=1}^n$ is a valid match, then there exists a valid match between $\{s_k\}_{k=1}^n$ and $\{s'_k\}_{k=1}^n$ that is order-preserving.

Proof:

As illustrated in Fig. 3, 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 $\{s'_k\}_{k=1}^n$ into $s''_1 \le s''_2 \le \ldots \le s''_n$. The match $\{(s_k,\ s''_k)\}_{k=1}^n$ is valid and order-preserving.

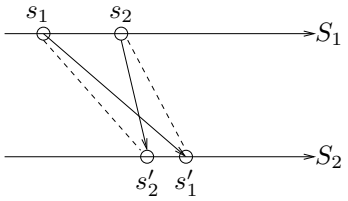

Fig. 3. More than one valid match: both the solid and the dotted lines are valid matches.

■

By Proposition 4.3, it suffices to consider only the matches that preserve the order of packets, and the problem is reduced to finding the departure that corresponds to the first arrival. With this idea in mind, we develop the following detector:

$$\delta_{\text{DM}}(S_1,\ S_2,\ \Delta,\ n) = \begin{cases} 1 & \text{if } \exists m \in [h_2(s_1^{(1)}),\ h_2(\Delta)] \text{ s.t. the match} \\ & \{(s_k^{(1)},\ s_{k+m-1}^{(2)})\}_{k=1}^n \text{ is valid,} \\ 0 & \text{o.w.} \end{cases}$$

which is implemented by the algorithm DM as shown in Table II.

TABLE II
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;
```

To analyze the complexity of DM, note that the inner loop has $O(n)$ operations, and the number of such loops is at most 1 plus the number of arrivals in the interval $[s_1^{(1)},\ \Delta)$ in S_2. Thus the complexity of DM is at most

$$n((\#\text{ arrivals in } [s_1^{(1)},\ \Delta) \text{ in } S_2) + 1).$$

[5]For example, if there are at most L departures during time Δ, then the exhaustive search for a length-n valid match has complexity $O(L^n)$.

Now we analyze the performance of DM. We will show that any stepping-stone pairs with bounded delay Δ will be detected by δ_{DM}, *i.e.,* there is no miss. We have shown by Proposition 4.3 that a stepping-stone pair with bounded delay Δ must have an order-preserving, valid match, and the problem of finding a valid match is reduced to a simpler problem of finding $s_m^{(2)}$ (*i.e.,* the match of $s_1^{(1)}$). There are some constraints on the range of $s_m^{(2)}$. The first constraint is causality, which requires $s_m^{(2)} \ge s_1^{(1)}$. The second is bounded delay. Since the monitor may not have started recording from the beginning of the streams, there may be packets sent before the monitor starts and received afterwards. This phenomenon, however, can only occur during time $[0,\ \Delta)$ because of the bounded delay assumption. Thus for any stepping-stone pair, $s_{h_2(\Delta)}^{(2)}$ has to be the relay of $s_k^{(1)}$ for some $k \ge 1$. As a result, m has to satisfy $h_2(s_1^{(1)}) \le m \le h_2(\Delta)$, as shown in Fig. 4. Therefore, DM must be able to return "ATTACK".

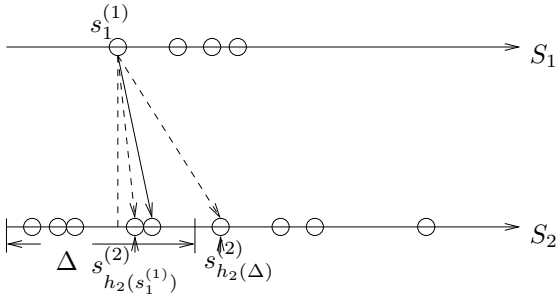

Fig. 4. The match of $s_1^{(1)}$: there are three possible candidates.

Next, we show that for independent Poisson normal pairs, the false alarm probability goes to zero exponentially, as stated in the following theorem.

Theorem 4.4: If S_1 and S_2 are independent Poisson processes of rates λ_1 and λ_2, respectively, then the false alarm probability of DM is bounded by

$$P_F(\delta_{\text{DM}}) \le \gamma^{n-1},$$

where $\gamma = 1 - e^{-\lambda_1\lambda_2\Delta/(\lambda_1+\lambda_2)}$.

Proof: See Appendix. ■

Remark: Theorem 4.4 gives a few insights into the problem. Since $\gamma \le 1 - e^{-\min(\lambda_1,\ \lambda_2)\Delta}$, we have $\gamma \to 0$ if $\min(\lambda_1,\ \lambda_2) \to 0$, *i.e.,* DM almost never falsely accuses slow independent Poisson traffic.

Intuitively, it is easier to match two processes of equal rates. This intuition is strengthened by Theorem 4.4 because $\gamma \le 1 - e^{-\lambda\Delta/2}$, where $\lambda = \max(\lambda_1,\ \lambda_2)$, and thus the upper bound for Poisson traffic of equal rates is larger.

Similar to DMV, we can also estimate the sample size required by DM to achieve a given false alarm probability α by calculating the value n that makes the upper bound in Theorem 4.4 equal to α, *i.e.,*

$$n = \log\alpha - \log\gamma + 1.$$

For example, if $\lambda_1 = \lambda_2 = 1$, and $\Delta = 10$, then a match length 682 suffices to guarantee a false alarm probability bounded by 1%. Note that for this match length, DM needs up to $2n + \lambda_2\Delta = 1374$ packets on the average to find a valid match.

V. Detecting Stepping-stone Pairs in Chaff

So far we have not considered how to detect stepping-stone traffic in chaff. In practice, the attacker usually combines traffic perturbation with the insertion of chaff to evade detection. In this section, we address how to detect bounded memory or bounded delay stepping-stone pairs in the presence of chaff. The key is to allow a limited violation of constraints so that the detection scheme can still distinguish stepping-stone pairs and normal pairs while tolerating certain amount of chaff.

A. The Bounded Memory Case

As explained in Section III, the maximum variation of stepping-stone pairs with bounded memory M is always bounded by M. After inserting chaff, the attacker can make the maximum variation larger than M. But with a limited number of chaff packets, the maximum variation will still be much less than that of independent processes. Based on this idea, we propose an algorithm called "DETECT-BOUNDED-MEMORY-CHAFF" (DBMC) presented in Table III. Algorithm DBMC has complexity $O(n)$.

TABLE III
DETECT-BOUNDED-MEMORY-CHAFF (DBMC).

DETECT-BOUNDED-MEMORY-CHAFF(S_1, S_2, M, n):

$S = \text{merge}(S_1, S_2)$;
$d = d_{\max} = d_{\min} = 0$;
$C = 0$;
for $w = 1 : n$
  if ($d_{\max} - d_{\min} = M$) and (($d = d_{\max}$, $s_w \in \mathcal{T}_1$) or ($d = d_{\min}$, $s_w \in \mathcal{T}_2$))
    $C = C + 1$;
  else
    $d = \begin{cases} d+1 & \text{if } s_w \in \mathcal{T}_1, \\ d-1 & \text{if } s_w \in \mathcal{T}_2; \end{cases}$
    $d_{\max} = \max(d_{\max}, d)$;
    $d_{\min} = \min(d_{\min}, d)$;
  end
end
return $\begin{cases} \text{ATTACK} & \text{if } \frac{C}{n} < \frac{1}{M+1}, \\ \text{NORMAL} & \text{o.w.}; \end{cases}$

If (S_1, S_2) is a pair of stepping-stone streams passing through a host with memory size M, the counter C in Table III counts the number of times the memory would have been underflowed ($d = d_{\min}$, $s_w \in \mathcal{T}_2$) or overflowed ($d = d_{\max}$, $s_w \in \mathcal{T}_1$) if chaff had not been inserted. Algorithm DBMC makes detection if the fraction (C/n) is suspiciously small. Since no attacking packet can violate the memory constraint (only chaff packets can), the number of chaff packets is at least C. Therefore, to evade DBMC, the attacker has to insert at least $n/(M+1)$ chaff packets in every n packets. We conclude that DBMC is robust against up to $1/(M+1)$ fraction of chaff.

It is difficult to characterize the false alarm probability of DBMC in closed form for finite n. As n increases, however, it can be shown that the false alarm probability goes to zero if normal pairs consist of independent Poisson processes; see [13].

B. The Bounded Delay Case

For stepping-stone pairs with bounded delay, we can always match the incoming packets with the outgoing packets (perhaps except for the first few outgoing packets) so that all matched pairs satisfy causality and bounded delay. When chaff is inserted, we may not be able to match all the packets. If the attacker does not insert enough chaff, however, the attacking traffic will have much more matched packets than normal traffic. The algorithm is presented in Table IV. Algorithm DBDC has complexity $O(n)$.

TABLE IV
DETECT-BOUNDED-DELAY-CHAFF (DBDC).

DETECT-BOUNDED-DELAY-CHAFF(S_1, S_2, Δ, n, λ):

$i = j = 1$;
$C = 0$;
while $i + j \le n$
  if $s_j^{(2)} - s_i^{(1)} < 0$
    $C = C + 1$; $j = j + 1$;
  else if $s_j^{(2)} - s_i^{(1)} > \Delta$
    $C = C + 1$; $i = i + 1$;
  else
    $i = i + 1$; $j = j + 1$;
  end
end
return $\begin{cases} \text{ATTACK} & \text{if } \frac{C}{n} < \frac{1}{1+\lambda\Delta}, \\ \text{NORMAL} & \text{o.w.}; \end{cases}$

Algorithm DBDC is inspired by an optimal chaff-inserting algorithm called "BOUNDED-GREEDY-MATCH" (BGM) proposed by Blum *et al.* [8]. For every arrival at time t, BGM matches it with the first unmatched departure in $[t,\ t+\Delta]$; if there is no departure in this interval or all the departures have been matched, BGM inserts a chaff packet at arrival t; BGM also inserts chaff at all the departures which have no arrivals to match to. Algorithm DBDC uses a counter C to record the number of chaff packets which would have been inserted had BGM been used, and reports alarm if the fraction (C/n) is smaller than a predetermined value. It is shown in [8] that BGM inserts the minimum chaff to embed a stepping-stone pair with bounded delay Δ into arbitrary point processes[6]. If the attacker wants to send attacking packets through a host with delays bounded by Δ, he needs to insert at least $n/(1+\lambda\Delta)$ chaff packets in every n packets to evade DBDC. Therefore, DBDC is robust against up to $1/(1+\lambda\Delta)$ fraction of chaff.

It can be shown (see [13]) that, for independent Poisson processes of rates bounded by λ, the false alarm probability of DBDC goes to zero as n goes to infinity. The value λ is a design parameter representing the tradeoff between robustness and false alarm probability. A smaller λ would allow DBDC to tolerate more chaff, but a larger λ would enable DBDC to have vanishing false alarm probability for a wider range of normal traffic.

[6]The original proof in [8] is for independent binomial processes, but it holds for arbitrary processes.

VI. Comparing the Algorithms

We have introduced techniques for detecting stepping-stone pairs with various constrained perturbations or a combination of perturbation and chaff. In practice, stepping-stone pairs may vary in what conditions they satisfy depending on the nature of the attacks. For certain types of stepping-stone pairs, more than one detection algorithm are applicable. The question is how to compare the performance of different algorithms in detecting such stepping-stone pairs.

A. Algorithms for Packet-conserving Transformations

1) DMV vs. DA: Blum *et al.* [8] consider the detection of stepping-stone pairs that satisfy both the bounded delay and the bounded peak rate conditions. The underlying idea is that in interactive stepping-stone attacks, usually not only is the delay bounded, but the peak rate at which the attacker can issue packets is also bounded because he needs time to process the feedback and type new commands. Specifically, Blum *et al.* consider stepping-stone pairs with bounded delay Δ, and the maximum number of arrivals within time t is $L(t)$. The second condition, referred to as the bounded peak rate condition, is formalized in the following definition:

Definition 6.1: A stepping-stone pair (S_1, S_2) has *bounded peak rate* $L(\cdot)$ if $\sup_r |\{s \in \mathcal{T}_1 : s \in [r, r+t]\}| \le L(t)$ for all $t \ge 0$.

Let $M \triangleq L(\Delta)$ be the largest number of packets the attacker can send during the maximum delay Δ. Note that stepping-stone pairs with bounded delay and bounded peak rate always use bounded memory, as stated in the following proposition:

Proposition 6.2: Define $N_i(a, b)$ be the number of packets on S_i in an interval $[a, b]$ $(a \le b)$. For a stepping-stone pair with bounded delay and bounded peak rate, if Δ is the maximum delay, and M is the maximum number of packets that the attacker can send within time Δ, then

$$|N_1(a, b) - N_2(a, b)| \le M, \quad \forall a \le b,$$

i.e., the stepping-stone pair uses bounded memory M.

Proof: See Appendix. ∎

By Proposition 6.2, we conclude that stepping-stone pairs with bounded delay and bounded peak rate are also stepping-stone pairs with bounded delay and bounded memory. Note that the inverse is not true, *i.e.,* bounded delay and bounded memory do not imply bounded peak rate.

Blum *et al.* in [8] propose a detection algorithm called "DETECT-ATTACKS" (DA) to detect stepping-stone pairs with bounded delay and bounded peak rate. Algorithm DA divides samples in $S_1 \cup S_2$ into groups of size $2(M+1)^2$. For each group, it computes the cumulative difference in that group. Then DA returns "NORMAL" if there exists a group with cumulative difference greater than M. The detector using DA is defined below:

$$\delta_{\text{DA}}(S_1, S_2, M, n) = \prod_{k=1}^{n/(2(M+1)^2)} \delta_{\text{DA}}^{(k)}(S_1, S_2, M),$$

where

$$\delta_{\text{DA}}^{(k)}(S_1, S_2, M) = \begin{cases} 1 & \text{if } \max\limits_{1 \le w \le 2(M+1)^2} |d^{(k)}(w)| \le M, \\ 0 & \text{o.w.,} \end{cases}$$

where $d^{(k)}(w)$ $(w = 1, \ldots, 2(M+1)^2)$ is the cumulative difference for packets in the kth group.

Blum *et al.* show that DA has no miss for stepping-stone pairs with bounded delay and bounded peak rate. Moreover, they prove that $2(M+1)^2 \log \frac{1}{\alpha}$ packets are needed to guarantee a false alarm probability no more than α.

We, however, are interested in the asymptotic behavior of DA in terms of error exponent. Note that [8] does not compute the error exponent for the false alarm probability of DA. To obtain its error exponent, we introduce the following lemma:

Lemma 6.3: For independent Poisson normal pairs,

$$\Pr\{\max_{i \in \{1, \ldots, m\}} |d(i)| \le M\} \le \frac{\sigma^m}{1-\sigma},$$

and when m is large enough,

$$\Pr\{\max_{i \in \{1, \ldots, m\}} |d(i)| \le M\} \ge K\sigma^m,$$

where $\sigma = \cos \frac{\pi}{2(M+1)}$, and $K = \frac{\sin \frac{\pi}{2(M+1)}}{2(M+1)(1-\sigma)}$.

Proof: See Appendix. ∎

If M is large, we can apply Lemma 6.3 to each group of $2(M+1)^2$ samples to obtain the upper and lower bounds on the false alarm probability of that group. Note that in [8] it is proved that the single group false alarm probability is upper bounded by $\frac{1}{2}$. Hence the false alarm probability of one group is upper bounded by

$$\min\left(\frac{\sigma^{2(M+1)^2}}{1-\sigma}, \frac{1}{2}\right) = \begin{cases} \frac{2+\sqrt{2}}{16} & \text{if } M = 1, \\ \frac{1}{2} & \text{if } M \ge 2. \end{cases}$$

Algorithm DA has a false alarm if all the $n/\left[2(M+1)^2\right]$ groups have false alarms[7], so for large M, the total false alarm probability satisfies

$$\left(K\frac{1}{2(M+1)^2}\sigma\right)^n \le P_F(\delta_{\text{DA}}) \le \left(\frac{1}{2}\right)^{\frac{n}{2(M+1)^2}}. \qquad (2)$$

Therefore, for large M, the false-alarm error exponent of DA is at most $-\log\left(K\frac{1}{2(M+1)^2}\sigma\right)$ and at least $\log 2/(2(M+1)^2)$.

We want to compare DA with DMV in detecting stepping-stone pairs with bounded delay and bounded peak rate. By Proposition 6.2, we have shown that such stepping-stone pairs satisfy the bounded memory condition. Thus DMV also has no miss. We now compare their false alarm probabilities.

We first point out that DMV always outperforms DA for any realization. One reason is that $v(w) \ge \max\limits_{1 \le i \le w} |d(i)|$ (see Fig. 5), and another is that DA restarts computation from $d^{(k)}(0) = 0$ at the beginning of each group, whereas DMV keeps increasing the maximum variation $v(w)$ across groups. Therefore, for every realization, if DMV has a false alarm, DA must have a false alarm too.

[7] In DA, the sample size n is always a multiple of $2(M+1)^2$.

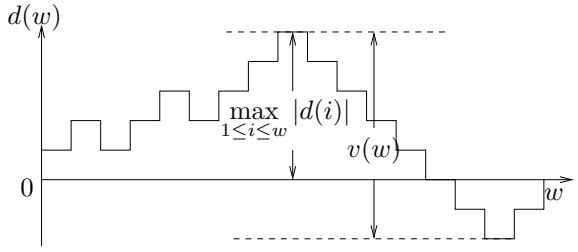

Fig. 5. The statistics used by DA and DMV.

Next we compare their false alarm probabilities. In particular, we are interested in whose false alarm probability has a larger error exponent. From Theorem 3.2 and (2), we see that the false-alarm error exponent of DMV is $-\log\rho$ whereas that of DA is at most $-\log\left(K^{\frac{1}{2(M+1)^2}}\sigma\right)$. By Taylor expansion of the error exponents, we have that as $M\to\infty$,

$$\begin{aligned} -\log\rho &= \frac{\pi^2}{2(M+2)^2}+o\left(\frac{1}{M^2}\right), \\ -\log\left(K^{\frac{1}{2(M+1)^2}}\sigma\right) &= \frac{\frac{\pi^2}{4}+\log\frac{\pi}{2}}{2(M+1)^2}+o\left(\frac{1}{M^2}\right). \end{aligned}$$

Therefore, for large M, the false-alarm error exponent of DMV is at least 3.38 times larger than that of DA.

2) DM vs. DMV: For stepping-stone pairs with both bounded memory and bounded delay, both DMV and DM can be used. We are interested in which algorithm performs better; particularly, we want to compare their asymptotic performance. Note that we need to give DMV and DM the same sample size to make a fair comparison. If we define sample size as the total number of monitored packets in both the incoming and the outgoing streams, then the sample size required by DM to find a length-n match is random; it is at most $(\#\text{ departures in }[0,\ \Delta))+2n$. For large n, the sample size is approximately $2n$. Hence we should compare $\delta_{\text{DMV}}(S_1,\ S_2,\ M,\ 2n)$ with $\delta_{\text{DM}}(S_1,\ S_2,\ \Delta,\ n)$.

Theorems 3.2 and 4.4 suggest that for Poisson processes of equal rates λ, DM is preferable if $\gamma\le\rho^2$, *i.e.,*

$$\lambda\le-\frac{4}{\Delta}\log\left(\sin\frac{\pi}{M+2}\right). \tag{3}$$

Otherwise, DMV is preferable. For example, for $M=40$, and $\Delta=10$, the threshold is $\lambda\le 1.0375$. This threshold phenomenon has an intuitive explanation. Algorithm DMV only uses the rank statistics, so it does not depend on the rate of the traffic; on the other hand, DM performs better on slower traffic and worse on faster traffic. The reason for the latter is that $\lambda\to 0$ means the inter-arrival time $\to\infty$, which is equivalent to having finite inter-arrival time but $\Delta\to 0$, *i.e.,* for extremely slow traffic, almost perfect synchrony is required to raise an alarm, and thus it is unlikely for DM to have false alarms. Similarly, if $\lambda\to\infty$, the inter-arrival time $\to 0$; equivalently, it means having non-zero inter-arrival time but $\Delta\to\infty$, *i.e.,* the delay constraint is essentially removed, which causes DM to always raise alarms. Therefore, when the traffic is sufficiently slow, DM outperforms DMV, and otherwise DMV performs better than DM. The comparison suggests that the bounded memory condition is more informative than the bounded delay condition in detecting stepping-stone traffic for $\lambda\Delta>4\log\left((M+2)/\pi\right)$. Since the right hand side merely grows as $\log M$, the memory bound can be advantageous even for modest rate and large memory. For example, for $M=10^6$ packets, $\Delta=10$ seconds, we only need $\lambda>5.1$ packets per second for the bound memory condition to provide better detection performance.

B. Algorithms Dealing with Chaff

In Section I-A, we have mentioned several existing detection schemes dealing with chaff evasion ([8]–[10]). We now compare these schemes to our proposed algorithms DBMC and DBDC.

1) DBMC vs. DAC: Detecting stepping-stone traffic with both bounded memory perturbation and chaff has not been addressed in the literature to the best of our knowledge. In [8], Blum *et al.* propose an algorithm called "DETECT-ATTACKS-CHAFF" (DAC) for detecting a more restricted class of stepping-stone traffic with bounded delay, bounded peak rate, and chaff. Algorithm DAC works exactly the same as DA except that the group size is increased from $2(M+1)^2$ to $8(M+1)^2$, and the threshold is increased from M to $2M$. It is shown in [8] that DAC is robust against up to M chaff packets in every $8(M+1)^2$ packets, and for independent Poisson traffic, the false alarm probability of DAC is bounded by $2^{-n/(8(M+1)^2)}$.

By Proposition 6.2, DBMC is also applicable to the stepping-stone traffic Blum *et al.* consider for DAC. We compare their performance in terms of robustness and false alarm probability. As stated in [8], the attacker can evade DAC by inserting $M+1$ chaff packets in a group of $8(M+1)^2$ packets. As the traffic size increases, the fraction of chaff needed to evade DAC becomes negligible. Algorithm DBMC, as argued in Section V-A, is robust against a number of chaff packets constituting $1/(M+1)$ fraction the total stepping-stone traffic. The drawback of DBMC is that its false alarm probability is only guaranteed asymptotically, whereas DAC has exponentially decaying false alarm probability.

2) DBDC vs. S-III [10]: For detecting stepping-stone traffic with both bounded delay and chaff, Peng *et al.* [9] and Zhang *et al.* [10] both provide partial solutions for the special case when chaff only appears in the outgoing traffic. Peng *et al.* [9] use a watermarking scheme which requires the detector to actively manipulate the traffic, and thus falls outside the scope of this paper. Zhang *et al.* [10] propose a scheme called "S-III" which matches every arrival at t_i with the first unmatched departure in $[t_i,\ t_i+\Delta]$, and makes detection if all the arrivals are successfully matched[8]. Scheme S-III is proved to have exponentially decaying false alarm probability for independent Poisson processes. If the attacker can insert chaff in the incoming traffic, however, one chaff packet is enough to defeat S-III. Algorithm DBDC, on the other hand,

[8]In [10], there is also a variation of S-III called "S-IV", which makes decision by comparing the minimum deviation among all the valid matches with a threshold. The false alarm probability of S-IV is no larger than that of S-III, but S-IV is also easy to be defeated by chaff in the incoming traffic.

is applicable to cases when both the incoming and outgoing streams are subject to chaff insertion. Furthermore, DBDC is robust against chaff traffic of non-zero rate. Its weakness, similar to DBMC, is that it does not have a guaranteed false alarm probability for finite sample size.

VII. Numerical Results

We simulate our algorithms on both synthetic data and traces to verify their performance. For synthetic data, we use independent Poisson processes of equal rates as our normal pairs; the goal of using synthetic data is to validate our analysis. For real data, we use the traces LBL-PKT-4, which contains an hour's worth of all wide-area traffic between the Lawrence Berkeley Laboratory and the rest of the world. The traces were made by Paxson and were first used in his paper [14].

A. Simulations on Synthetic Data

In this section, we simulate DA, DMV, and DM on synthetic independent Poisson processes to verify their false alarm probabilities. We let $M = 40$ packets, $\Delta = 10$ seconds, and vary the sample size between 2500 and 5000 packets (including both incoming and outgoing packets)[9]. The performance of DA and DMV does not depend on the traffic rate because they only rely on the relative order of packets. For DM, rate does play a significant role and will be specified when it is necessary.

We have shown the advantage of DMV over DA and have quantified their difference in terms of error exponent as $M \to \infty$ in Section VI-A.1. We now show how their performance compares for finite M. In Fig. 6, we plot the simulated false alarm probabilities of DMV and DA, together with the upper bound on $P_F(\delta_{\text{DMV}})$ from Theorem 3.2 and the asymptotic upper and lower bounds on $P_F(\delta_{\text{DA}})$ from (2). Simulation shows that the asymptotic bounds in (2) are valid even for relatively small M ($M = 40$). Furthermore, it confirms our claim that the false alarm probability of DMV decays much faster than that of DA.

We simulate DM for different traffic rates ($\lambda = 3,\ 3.5,\ 4,\ 4.5$). The simulation results are plotted in Fig. 7. The upper bounds in Theorem 4.4 for rates between 3 and 4.5 are close to 1; the actual false alarm probabilities obtained from simulation are much lower. The plot shows that the upper bound in Theorem 4.4 is not tight, but it correctly predicts the fact that $P_F(\delta_{\text{DM}})$ increases with the increase of traffic rate, as argued in Section VI-A.2.

Furthermore, we make an overall comparison by plotting the simulated false alarm probabilities of DA, DMV and DM together in Fig. 8. From the plot it is clear that the comparison between DM and DMV depends on the traffic rate. In our simulation, $M = 40$, $\Delta = 10$, the threshold rate estimated by (3) is about 1.0375. The simulation verifies the existence of such a threshold rate because the false alarm probability of DM decays faster than that of DMV for $\lambda = 3.5$ and slower for $\lambda = 4.5$. Note, however, that in the estimation of the threshold rate we are conservative about DM. This is because for DMV, Theorem 3.2 gives the exact error exponent, whereas for DM, Theorem 4.4 only characterizes a lower bound on its error exponent (which is shown to be not tight). Therefore, we expect that the actual threshold rate is larger than the one estimated by (3), *e.g.*, in the simulation the threshold rate is about 4 .

[9]Note that since DA requires the sample size to be a multiple of $2(M+1)^2 = 3362$ packets, we extend the sample size for DA to 6724.

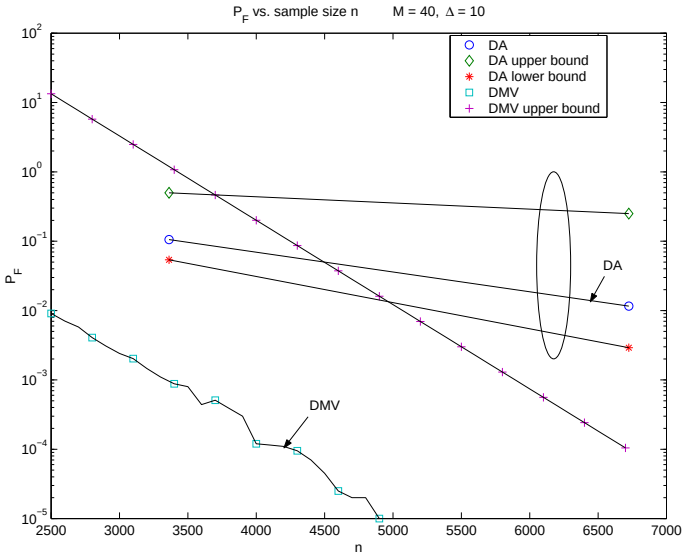

Fig. 6. $P_F(\delta_{\text{DA}})$, $P_F(\delta_{\text{DMV}})$, and their bounds; $M = 40$ packets, 100000 Monte Carlo runs.

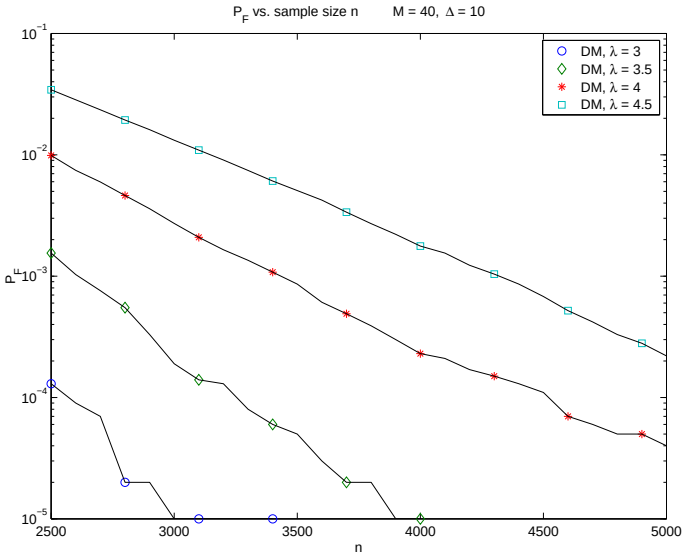

Fig. 7. $P_F(\delta_{\text{DM}})$ under various rates; $\Delta = 10$ seconds, 100000 Monte Carlo runs.

B. Simulations on Traces

We extract 134 flows from the TCP packets in LBL-PKT-4. Each flow has at least 1000 packets, and 4 of them have at least 10000 packets. For the testing of false alarm probabilities, we take all combinations of the 134 flows, filter out the pairs satisfying the definition of stepping-stone pairs with bounded

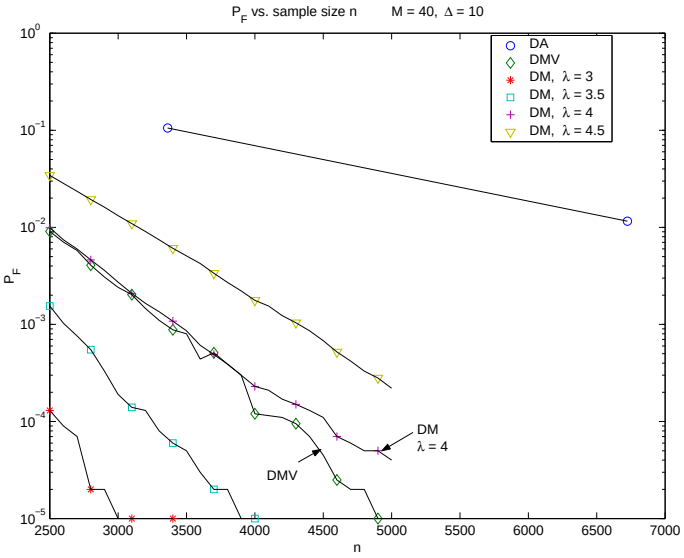

Fig. 8. $P_F(\delta_{\text{DA}})$, $P_F(\delta_{\text{DMV}})$, and $P_F(\delta_{\text{DM}})$; $M = 40$ packets, $\Delta = 10$ seconds, 100000 Monte Carlo runs.

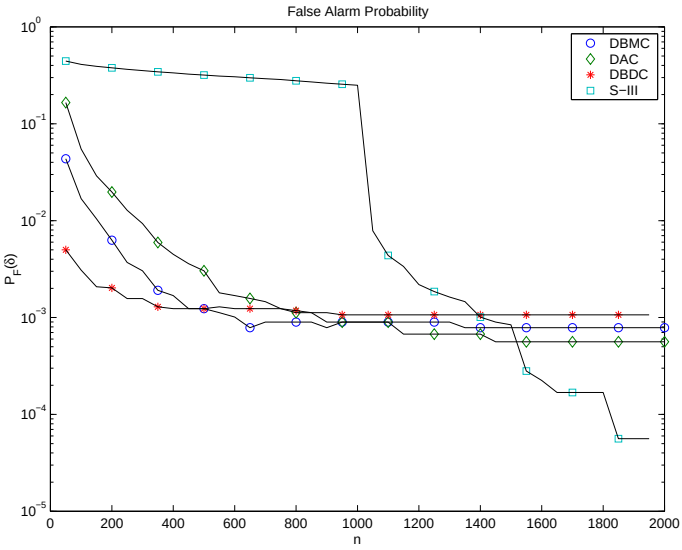

Fig. 9. $P_F(\delta_{\text{DBMC}})$, $P_F(\delta_{\text{DAC}})$, $P_F(\delta_{\text{DBDC}})$, and $P_F(\delta_{\text{S-III}})$ on LBL-PKT-4 (n is the joint sample size).

memory or bounded delay[10], and treat the rest as normal pairs. For the testing of miss probabilities, we introduce independent timing perturbation and chaff into the 4 flows with 10000 packets to generate independent copies of their stepping-stone relays. To generate bounded delay perturbation, we add to every packet a delay chosen independently and uniformly from $[0,\ \Delta]$. To generate bounded memory perturbation, we divide packets into segments of size $M/2$, and randomly generate $M/2$ relay packets in the $(i+1)$th segment for $i = 1,\ 2, \ldots$ [11]. Furthermore, we insert N_c chaff packets in both S_1 and S_2 according to uniform distributions on the range of the flows. In this section, we let $M = 20$ packets, $\Delta = 5$ seconds, and $N_c = 1000$ packets. In DBDC, we also set $\lambda = 2.6$ packets per second.

We first simulate the false alarm probabilities of DBMC, DAC, DBDC, and S-III; see Fig. 9. The false alarm probabilities of DBMC, DAC, and DBDC are comparable, and they do not change much after sample size 1000; the false alarm probability of S-III, however, keeps decreasing after 1000 packets to a much smaller value. From the plot, we see that the false alarm probabilities of DBMC, DAC, and DBDC for the traces do not decay exponentially. It is possible that the false alarm probability of S-III still decays exponentially, but we do not have enough data in these traces to verify that.

We then simulate the miss probabilities of DBMC and DAC on the long flows (of size 10000) and their synthetic relays (Fig. 10). For each of the 4 long flows, we generate 1000 independent relay flows by random segment generation and uniform chaff insertion. Thus we totally have 4000 stepping-stone pairs with bounded memory in chaff. The simulation shows that DBMC has much lower miss probability than DAC.

[10]The filtering is done by running DMV or DM on the entire flow pairs, and excluding the pairs reported as "ATTACK".

[11]The departures in the $(i+1)$th segment can be viewed as relays of the packets arriving in the ith segment. It is easy to see that such perturbation satisfies the bounded memory constraint.

In fact, DBMC detects all the stepping-stone pairs in our simulation, whereas DAC has up to 27.58% miss by sample size 22000. The plot also shows that the miss probability of DAC increases with the increase of the average number of chaff packets. This result conforms to our analysis because the number of chaff packets that are needed to evade DBMC is proportional to the traffic size, whereas DAC can be evaded by a fixed number of chaff packets. Note that our robustness claim about DBMC is conservative; DBMC is robust against up to $1/(M+1) \approx 0.0476$ fraction of chaff no matter how the chaff packets are inserted. In the simulation, DBMC survives 0.1 fraction of chaff, which implies that the uniform chaff insertion is not optimal for bounded memory stepping-stone pairs.

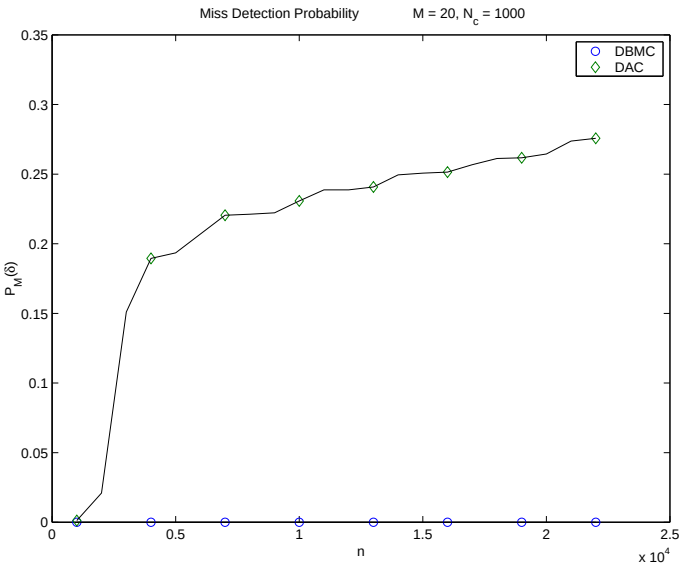

Fig. 10. $P_M(\delta_{\text{DBMC}})$ and $P_M(\delta_{\text{DAC}})$ (n is the joint sample size).

We next simulate the miss probabilities of DBDC and S-

III; see Fig. 11. For each of the 4 long flows, we generate 1000 independent relay flows by introducing independent delays and chaff packets. The plot confirms that DBDC has a much smaller miss probability than S-III; actually, in the simulation, DBDC has no miss for almost all the sample sizes[12]. This is expected because DBDC is robust against up to a certain fraction of chaff packets no matter where they are inserted, whereas S-III is vulnerable to the chaff insertion into S_1. We see that in the simulation DBDC successfully handles 0.1 fraction of chaff, which is larger than the fraction $1/(1+\lambda\Delta) \approx 0.0714$ which DBDC is guaranteed to be able to handle. Similar to the case of DBMC, this shows that the uniform chaff insertion is not optimal for bounded delay stepping-stone pairs, either. From Fig. 10 and Fig. 11, we see that DAC is more robust to chaff than S-III.

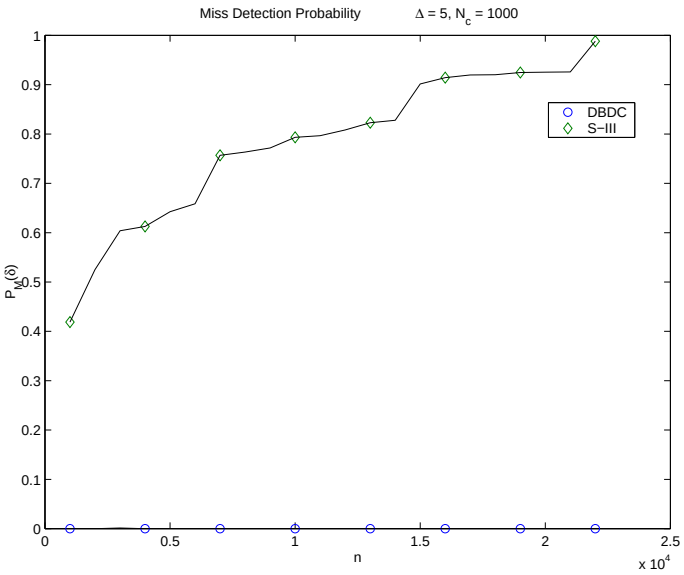

Fig. 11. $P_M(\delta_{\text{DBDC}})$ and $P_M(\delta_{\text{S-III}})$ (n is the joint sample size).

VIII. Conclusion

In this paper, we develop techniques to detect encrypted stepping-stone connections. These techniques can rule out independent connection pairs and leave a much smaller number of suspicious connections for further examination. After discovering all the stepping-stone connections, one can use existing serialization methods to determine the intrusion path [15].

APPENDIX

Proof of Theorem 3.2 and Lemma 6.3

The proof is based on the theory of random walk. Let $\{X_n\}_{n\geq 0}$ be a simple random walk, *i.e.*,

$$X_0 = 0, \quad X_n = Z_1 + Z_2 + \ldots + Z_n, \ (n > 0)$$

[12]It is except for the sample size 3000, where we have 6 misses out of 4000 Monte Carlo runs.

where $\{Z_i\}_{i=1,\,2,\ldots}$ are i.i.d. random variables taking value in $\{-1,\,0,\,1\}$. Let $p = \Pr\{Z_i = 1\}$, $q = \Pr\{Z_i = -1\}$. Define *the hitting time* of $-b$ or a ($a,\ b \geq 0$) as

$$N_{-b,\,a} = \inf\{n \geq 1 :\ X_n = -b \text{ or } a\}. \tag{4}$$

The following lemma is from [16]:

Lemma 8.1:

$$\Pr\{N_{-b,\,a} = n\} \leq \frac{1}{2}\left(\frac{p}{q}\right)^{a/2} \frac{1}{s_1^{n-1}} + \frac{1}{2}\left(\frac{q}{p}\right)^{b/2} \frac{1}{s_1^{n-1}}, \tag{5}$$

where $s_1 = \frac{1}{1-p-q+2(pq)^{\frac{1}{2}}\cos\left(\frac{\pi}{a+b}\right)}$. If $a = b$, then for large n,

$$\Pr\{N_{-b,\,a} = n\} \geq \frac{\sin\frac{\pi}{2a}}{2a s_1^{n-1}}. \tag{6}$$

Moreover, there exist constants c_v $(v = 1, \ldots, a+b-1)$ and s_v $(v = 2, \ldots, a+b-1)$ not depending on n, *s.t.*

$$\Pr\{N_{-b,\,a} > n\} = \sum_{v=1}^{a+b-1} \frac{c_v}{s_v^n} \tag{7}$$

where $|s_1| \leq |s_v|$ $(v = 2, \ldots, a+b-1)$.

Since

$$\Pr\{N_{-b,\,a} > n\} = \sum_{r=n+1}^{\infty} \Pr\{N_{-b,\,a} = r\},$$

(5,6) give upper and lower bounds on $\Pr\{N_{-b,\,a} > n\}$.

For the proof of Theorem 3.2, note that for independent Poisson processes, it is known that $d(w)$ is a simple random walk. Define extreme values $U_n = \max_{i=0,\ldots,\,n} d(i)$, $L_n = \min_{i=0,\ldots,\,n} d(i)$. A false alarm occurs in DMV if and only if $U_n - L_n < M+1$. Note that the false alarm probability is the largest if $d(w)$ is symmetric (*i.e.*, $p = q = \frac{1}{2}$). Then we have

$$\begin{aligned} P_F(\delta_{\text{DMV}}) &= \Pr\{U_n - L_n < M+1\} \\ &= \Pr\{\bigcup_{a=1}^{M+1} \{U_n < a,\ L_n > -(M+2-a)\}\} \\ &\leq \sum_{a=1}^{M+1} \Pr\{U_n < a,\ L_n > -(M+2-a)\} \quad (8) \\ &\leq (M+1)\frac{\rho^n}{1-\rho}, \quad (9) \end{aligned}$$

where $\rho = \cos\frac{\pi}{M+2}$. Here (8) is by union bound, and (9) is by noticing

$$\Pr\{U_n < a,\, L_n > -(M+2-a)\} = \Pr\{N_{-(M+2-a),\,a} > n\},$$

and then applying (5) with $p = q = \frac{1}{2}$. Furthermore, by (7) it is easy to see that

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

For the proof of Lemma 6.3, note that

$$\Pr\{\max_{i\in\{1,\ldots,\,n\}} |d(i)| \leq M\} = \Pr\{N_{-(M+1),\,(M+1)} > n\}.$$

Applying (5, 6) with $a = b = M+1$ and $p = q = \frac{1}{2}$ gives the desired result.

■

Proof of Theorem 4.4

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 $i = 1, \ldots,\, n$.

For $i \ge 2$, define the interarrival times be $U_i \triangleq s_i - s_{i-1}$, and $V_i \triangleq s_i' - s_{i-1}'$. Let $Z_i \triangleq V_i - U_i$. Then

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

Therefore, given Y_1, $\{Y_i\}_{i=2}^{\infty}$ is a general random walk with steps Z_i's. We know that V_i and U_i are independent Exponential random variables with mean $1/\lambda_2$ and $1/\lambda_1$, respectively, and thus Z_i's are i.i.d. with distribution function

$$\begin{aligned}\Pr\{Z_i \le z\} &= \Pr\{V_i - U_i \le z\} \\ &= \int_{\max(0,\,-z)}^{\infty} p_{U_i}(u)\Pr\{V_i \le u + z\}du \\ &= \begin{cases} 1 - \frac{\lambda_1}{\lambda_1+\lambda_2}e^{-\lambda_2 z} & \text{if } z \ge 0, \\ \frac{\lambda_2}{\lambda_1+\lambda_2}e^{\lambda_1 z} & \text{if } z < 0. \end{cases}\end{aligned}$$

The probability density function (pdf) of Z_i is

$$p_Z(z) = \begin{cases} \frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2}e^{-\lambda_2 z} & \text{if } z \ge 0, \\ \frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2}e^{\lambda_1 z} & \text{if } z < 0. \end{cases}$$

The false alarm probability satisfies

$$\begin{aligned}P_F(\delta_{\text{DM}}) &= \Pr\{\exists s_1',\ \textit{s.t.}\ 0 \le Y_1^n \le \Delta\} \\ &\le \max_{y_1 \in [0,\,\Delta]} \Pr\{0 \le Y_2^n \le \Delta | Y_1 = y_1\}.\end{aligned}$$

Fix a $y_1 \in [0,\, \Delta]$. For $n \ge 2$, define

$$p_n(z)dz \triangleq \Pr\{Y_2^{n-1} \in [0,\, \Delta],\ z < Y_n < z + dz | Y_1 = y_1\}.$$

Define $p_1(z) = \delta(z - y_1)$ (Dirac delta function). In [16] (page 53) it is shown that

$$p_n(z) = \int_0^{\Delta} p_{n-1}(x)p_Z(z-x)dx.\ (n = 2,\, 3, \ldots)$$

Then we have

$$\begin{aligned}\Pr\{0 \le Y_2^n \le \Delta | Y_1 = y_1\} &= \int_0^{\Delta} p_n(z_n)dz_n \\ &= \int_0^{\Delta} p_{n-1}(z_{n-1})dz_{n-1} \\ &\qquad \int_0^{\Delta} p_Z(z_n - z_{n-1})dz_n \\ &= \int_0^{\Delta} p_Z(z_2 - y_1)dz_2 \\ &\qquad \int_0^{\Delta} p_Z(z_3 - z_2)dz_3 \cdots \\ &\qquad\qquad \int_0^{\Delta} p_Z(z_n - z_{n-1})dz_n.\end{aligned}$$

Let $\gamma \triangleq \max\limits_{t \in [0,\,\Delta]} \int_{-t}^{\Delta - t} p_Z(z)dz$. Simple calculation yields that $\gamma = 1 - e^{-\lambda_1\lambda_2\Delta/(\lambda_1+\lambda_2)}$. Then

$$\Pr\{0 \le Y_2^n \le \Delta | Y_1 = y_1\} \le \gamma^{n-1}.$$

Since this is true for all $y_1 \in [0, \Delta]$, we have $P_F(\delta_{\text{DM}}) \le \gamma^{n-1}$. ■

Proof of Proposition 6.2

If $b - a \le \Delta$, then

$$|N_1(a,\, b) - N_2(a,\, b)| \le \max(N_1(a,\, b),\, N_2(a,\, b)) \le M.$$

For $b - a > \Delta$, let $N_1'(a - \Delta,\, a)$ be the number of packets that arrive in $[a - \Delta, a)$ and departure after a, and $N_1''(b - \Delta, b)$ be the number of packets that arrive in $(b - \Delta, b]$ and departure before b. Then

$$\begin{aligned}N_1(a,\, b) &= N_1(a,\, b - \Delta) + N_1(b - \Delta,\, b), \\ N_2(a,\, b) &= N_1(a,\, b - \Delta) + N_1'(a - \Delta,\, a) \\ &\qquad + N_1''(b - \Delta,\, b),\end{aligned}$$

We have

$$\begin{aligned}N_2(a,\, b) - N_1(a,\, b) &= N_1'(a - \Delta,\, a) + N_1''(b - \Delta,\, b) \\ &\qquad - N_1(b - \Delta,\, b).\end{aligned}$$

Since $N_1''(b - \Delta,\, b) \le N_1(b - \Delta,\, b)$ and $N_1'(a - \Delta,\, a) \le M$, we have

$$N_2(a,\, b) - N_1(a,\, b) \le N_1'(a - \Delta,\, a) \le M.$$

Since $N_1'(a - \Delta, a) \ge 0$, $N_1''(b - \Delta, b) \ge 0$ and $N_1(b - \Delta, b) \le M$, we have

$$N_2(a,\, b) - N_1(a,\, b) \ge -N_1(b - \Delta,\, b) \ge -M.$$

■

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. 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] P. Peng, P. Ning, D. Reeves, and X. Wang, "Active Timing-Based Correlation of Perturbed Traffic Flows with Chaff Packets," in *Proc. 25th IEEE International Conference on Distributed Computing Systems Workshops*, (Columbus, OH), pp. 107–113, June 2005.

[10] L. Zhang, A. Persaud, A. Johson, and Y. Guan, "Detection of Stepping Stone Attack under Delay and Chaff Perturbations," in *Proc. of the 25th IEEE International Performance Computing and Communications Conference (IPCCC 2006)*, (Phoenix, AZ), April 2006.

[11] T. He and L. Tong, "A Signal Processing Perspective to Stepping-stone Detection," in *Proc. 2006 Conference on Information Sciences and Systems*, (Princeton, NJ), March 2006.

[12] J. Giles and B. Hajek, "An Information-Theoretic and Game-Theoretic Study of Timing Channels," *IEEE Transactions on Information Theory*, vol. 48, pp. 2455–2477, September 2002.
[13] T. He and L. Tong, "Detecting Stepping-Stone Traffic in Chaff: Fundamental Limits and Robust Algorithms," Tech. Rep. ACSP-TR-06-06-01, Cornell University, June 2006. `http://acsp.ece.cornell.edu/pubR.html`.
[14] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, June 1995.
[15] X. Wang, "The loop fallacy and serialization in tracing intrusion connections through stepping stones," in *Proc. of the 2004 ACM Symposium on Applied Computing*, (Nicosia, Cyprus), pp. 404–411, March 2004.
[16] D. Cox and H. Miller, *The Theory of Stochastic Processes*. New York: John Wiley & Sons Inc., 1965.