Robust Correlation of Encrypted Attack Traffic Through Stepping Stones by Manipulation of Interpacket Delays

Xinyuan Wang
Department of Computer Science
N.C. State University
Raleigh, NC 27695
xwang5@unity.ncsu.edu

Douglas S. Reeves
Cyber Defense Lab
Departments of Computer Science and
Electrical and Computer Engineering
N.C. State University
Raleigh, NC 27695
reeves@csc.ncsu.edu

ABSTRACT

Network based intruders seldom attack directly from their own hosts, but rather stage their attacks through intermediate “stepping stones” to conceal their identity and origin. To identify attackers behind stepping stones, it is necessary to be able to correlate connections through stepping stones, even if those connections are encrypted or perturbed by the intruder to prevent traceability.

The timing-based approach is the most capable and promising current method for correlating encrypted connections. However, previous timing-based approaches are vulnerable to packet timing perturbations introduced by the attacker at stepping stones.

In this paper, we propose a novel watermark-based correlation scheme that is designed specifically to be robust against timing perturbations. The watermark is introduced by slightly adjusting the timing of selected packets of the flow. By utilizing redundancy techniques, we have developed a robust watermark correlation framework that reveals a rather surprising result on the inherent limits of independent and identically distributed (*iid*) random timing perturbations over sufficiently long flows. We also identify the tradeoffs between timing perturbation characteristics and achievable correlation effectiveness. Experiments show that the new method performs significantly better than existing, passive, timing-based correlation in the presence of random packet timing perturbations.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General –*Security and protection (e.g., firewalls)*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – *Unauthorized access (e.g., hacking, phreaking)*.

General Terms

Security, Reliability

Keywords

Stepping Stones, Intrusion Tracing, Correlation, Robustness

1. INTRODUCTION

Network based attacks have become a serious threat to the critical information infrastructure on which we depend. Those charged with defending networked assets that are under attack would like very much to be able to identify the source of the attack, so that appropriate action can be taken (whether that be contacting the source network administrator, filtering the attacker’s traffic, litigation, or criminal prosecution). Attackers, however, go to some lengths to conceal their identities, using a variety of countermeasures. As an example, they may spoof the IP source address of their traffic. Methods of tracing spoofed traffic, generically referred to as IP traceback[6,11,13], have been developed to address this countermeasure.

Another common and effective countermeasure used by network-based intruders to hide the origin of their traffic is to connect through a sequence of stepping stones[14,15,20] before attacking the final target. For example, an attacker at host A may Telnet or SSH into host B, and from there launch an attack on host C. In effect, incoming packets of an attack connection or flow from A to B are forwarded by B, and become outgoing packets of a connection from B to C. The two connections are said to be related in this case. The victim at host C can use IP traceback to determine the attack comes from host B, but traceback will not be able to determine the attack originated from host A. To trace attacks through a stepping stone, it is necessary to correlate incoming traffic at the stepping stone with outgoing traffic at the stepping stone. This would allow the attack to be traced back to host A in the example.

The earliest work on connection correlation was based on tracking users’ login activities at different hosts [7,12]. Later work relied on comparing the packet contents, or payloads, of the connections to be correlated [14,17]. Most recent work has focused on the timing characteristics [16,19,20] of connections, in order to correlate encrypted connections (i.e. traffic encrypted using IPSEC[8] or SSH [10,18]).

Existing timing-based correlation approaches, are vulnerable to countermeasures by the attacker. In particular, the attacker can

CCS’03, October 27–30, 2003, Washington, DC, USA.

This work was supported by DARPA contract F30602-99-1-0540. The views and conclusion contained herein are those of the authors.

perturb the timing characteristics of a connection by selectively or randomly introducing extra delays when forwarding packets at the stepping stone. This kind of timing perturbation will adversely affect the effectiveness of any timing-based correlation. The timing perturbation could either make unrelated flows have similar timing characteristics, or make related flows exhibit different timing characteristics. Either case could cause a timing-based correlation method to fail.

In this paper, we address the random timing perturbation problem in correlating encrypted connections through stepping stones. Our goal is to develop a practical correlation scheme that is robust against random timing perturbation, and to answer fundamental questions concerning the maximum effectiveness of such techniques, and the tradeoffs involved in implementing them.

We propose a novel watermark-based connection correlation method that is designed to be robust against random timing perturbations by the attacker. The idea is to actively embed some unique watermark into the flow by slightly adjusting the timing of selected packets in the flow. If the embedded watermark is unique enough and robust against timing perturbation by the attacker, the watermarked flow can be uniquely identified, and thus effectively correlated. By utilizing a redundant watermark, we have developed a robust correlation scheme which can achieve a detection (true positive) rate arbitrarily close to 100%, and a watermark collision (false positive) rate arbitrarily close to 0 at the same time. This can be accomplished for an arbitrarily large (but bounded) independent and identically distributed (iid) random timing perturbation of arbitrary distribution, with an arbitrarily small adjustment of inter-packet timing, as long as there are enough packets in the flow to be watermarked.

The contributions of this paper are as follows. First, we demonstrate that a previously-proposed passive, timing-based correlation scheme is vulnerable to random timing perturbation. Second, we develop a practical watermark-based correlation scheme that is much more robust in the presence of random timing perturbations. Our experimental results show that the new method consistently has a higher detection (true positive) rate, whether there is random timing perturbation or not. Third, we prove that it is possible to achieve arbitrarily close to 100% true positive correlation rate and arbitrarily close to 0% false positive correlation rate at the same time, at least in theory, for sufficiently long flows under certain conditions. Lastly, we develop accurate models of the tradeoffs between the desired watermark correlation true positive rate (and false positive rate) and the watermark embedding parameters, as well as the defining characteristics of the random timing perturbation. The quantitative expression of the tradeoffs is of significant practical importance in optimizing the overall correlation effectiveness under a range of conditions.

The remainder of this paper is organized as follows. Section 2 summarizes previous work. Section 3 overviews watermark-based correlation. Section 4 describes the basic embedding of a single watermark bit in the inter-packet timing domain. Section 5 presents a probabilistically-robust watermark bit embedding. Section 6 analyzes the watermark bit robustness and tradeoffs. Section 7 analyzes the overall watermark detection and watermark collision. Section 8 evaluates the correlation effectiveness of our method experimentally. Section 9 concludes the paper, and describes future research directions.

2. PREVIOUS RELATED WORK

Existing connection correlation approaches are based on three different characteristics: 1) host activity; 2) connection content (i.e., packet payloads); and 3) connection (packet) timing. The host activity approach (e.g., CIS[7] and DIDS[12]) collects and tracks user login activities at each stepping stone. The fundamental problem of host activity approaches is that the user login activity information collected from stepping stones is not trustworthy. Since the attacker is assumed to have full control over each stepping stone, the attacker can easily modify, delete, or forge local user login information. This defeats the ability to perform correlation based on host activity.

Approaches based on connection content (e.g., Thumbprinting[14] and SWT[17]) require that payload content be invariant across stepping stones. Since the attacker can encrypt the flows that pass through the stepping stones, and thus modify the connection contents, this approach is limited to unencrypted connections.

Connection timing based approaches (e.g., IPD-based[16], Deviation-based[19] and ON/OFF-based[20]) use the arrival and/or departure times of packets to correlate connections. For example, IPD-based correlation [16] has shown that 1) the inter-packet timing characteristics of connections are preserved across many router hops and stepping stones; and 2) the timing characteristics of telnet and SSH connections are almost always unique enough to provide correct correlation across stepping stones.

While timing-based correlation is currently the most capable and promising correlation approach, existing timing-based correlation schemes are vulnerable to the attacker's use of active timing perturbation. Donoho et al. [5] have recently investigated the theoretical limits on the attacker's ability to disguise his traffic through timing perturbation and packet padding (i.e., injection of bogus packets). They show that correlation from the long term behavior (of sufficiently long flows) is still possible despite certain timing perturbations by the attacker. However, they do not present any tradeoffs between the magnitude of the timing perturbation, the desired correlation effectiveness, and the number of packets needed. Another important issue that is not addressed by [5] is the correlation false positive rate. While the coarse scale analysis for long term behavior may filter out packet jitter introduced by the attacker, it could also filter out the inherent uniqueness and details of the flow timing. Therefore coarse scale analysis tends to increase the correlation false positive rate while increasing the correlation true positive rate of timing-perturbed connections. Nevertheless, Donoho et al.'s work [5] represents an important first step toward a better understanding of the inherent limitations of timing perturbation by the attacker on timing-based correlation. Issues that were not addressed included whether correlation is effective for arbitrarily distributed (rather than Pareto distribution conserving) random timing perturbation, and the achievable tradeoff of false and true positive rates.

In the following sections we investigate these and other issues.

3. OVERVIEW OF WATERMARK-BASED CORRELATION

The objective of watermark-based correlation is to make the correlation of encrypted connections robust against random timing perturbations introduced by the attacker. Unlike existing timing-based correlation schemes, our watermark-based correlation is "active" in that it embeds a unique watermark into encrypted

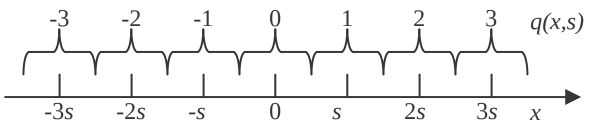

Figure 1. Quantization of the Scalar Value *x*

flows by slightly adjusting the timing of selected packets. The unique watermark that is embedded gives us an advantage over passive timing based correlation in resisting timing perturbation.

We assume the following about the random timing perturbation:

1) While the attacker can add extra delay to any or all packets of an outgoing flow of the stepping stone, the maximum delay he/she can introduce is bounded.
2) The random timing perturbation on each packet is independent and identically distributed (*iid*)
3) All packets in the original flow are kept in their original order, i.e., no padding packet is added and no packet is dropped by the attacker
4) While the watermarking scheme may be known to the attacker, the parameters of the watermarking are not known by the attacker.

3.1 Watermarking Model and Concept

Generally, digital watermarking[1] involves the selection of a watermark carrier domain and the design of two complementary processes: embedding and decoding. The watermark embedding process embeds the watermark bits into the carrier signal by a slight modification of some property of the watermark carrier, and the watermark decoder process detects and extracts any watermark bits (or equivalently determines the existence of a given watermark) from the carrier signal. To correlate encrypted connections, in this paper we propose to use inter-packet timing as the watermark carrier domain.

For a unidirectional flow of n>1 packets, we use t_i and t'_i to represent the arrival and departure times, respectively, of the *i*th packet P_i of a flow incoming to and outgoing from some stepping stone. (Given a bidirectional connection, we can split it into two unidirectional flows and process each independently.)

Assume without loss of generality that the normal processing and queuing delay added by the stepping stone is a constant c>0, and that the attacker introduces extra delay d_i to packet P_i at the stepping stone; then we have $t'_i = t_i + c + d_i$.

We define the *arrival inter-packet delay* (AIPD) between P_i and P_j as

$$ipd_{i,j} = t_j - t_i \qquad (1)$$

and the *departure inter-packet delay* (DIPD) between P_i and P_j as

$$ipd'_{i,j} = t'_j - t'_i \qquad (2)$$

We will use IPD to denote either AIPD or DIPD when it is clear in the context. We further define the *impact* or *perturbation* on $ipd_{i,j}$ by the attacker as the difference between $ipd'_{i,j}$ and $ipd_{i,j}$: $ipd'_{i,j} - ipd_{i,j} = d_j - d_i$.

Assume D>0 is the maximum delay that the attacker can add to P_i (i=1,…,n), then the impact or perturbation on $ipd_{i,j}$ is $d_j - d_i \in [-D, D]$. Accordingly range [-*D*, *D*] is called the *perturbation range* of the attacker.

To make our method robust against timing attacks, we choose to embed the watermark only over selected IPDs. The selection of IPDs requires randomly choosing the set of packets and random pairing of those chosen packets to get IPDs. The random IPD selection is unknown to the attacker; it should be difficult for the attacker to detect the existence of, extract, or corrupt the embedded watermark, without knowing the IPD selection function and other watermark embedding parameters.

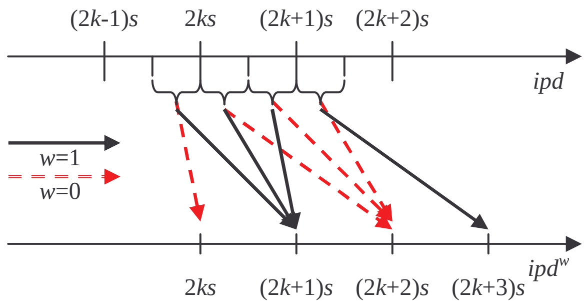

Figure 2. Mapping between Unwatermarked *ipd* and Watermarked *ipd*^{w} to Embed Watermark Bit *w*

4. EMBEDDING A SINGLE WATERMARK BIT INTO ONE IPD

4.1 Basic Watermark Bit Embedding and Decoding

As an IPD is conceptually a continuous value, we will first quantize the IPD before embedding the watermark bit. Given any IPD ipd>0, we define the *quantization of ipd* with uniform quantization step size s>0 as the function

$$q(ipd, s) = round(ipd / s) \qquad (3)$$

where round(x) is the function that rounds off real number x to its nearest integer (i.e., round(x) = i for any $x \in (i - ½, i + ½]$).

Figure 1 illustrates the quantization of scalar x. It is easy to see that $q(k \times s, s) = q(k \times s + y, s)$ for any integer k and any $y \in (-s/2, s/2]$.

Let *ipd* denote the original IPD before watermark bit *w* is embedded, and ipd^w denote the IPD after watermark bit *w* is embedded. To embed a binary bit *w* into an IPD, we slightly adjust that IPD such that the quantization of the adjusted IPD will have *w* as the remainder when the modulus 2 is taken.

Given any ipd>0, s>0 and binary bit *w*, the watermark bit embedding is defined as function

$$e(ipd, w, s) = [q(ipd + s/2, s) + \Delta] \times s \qquad (4)$$

where $\Delta = (w - (q(ipd + s/2, s) \bmod 2) + 2) \bmod 2$.

The embedding of one watermark bit *w* into scalar *ipd* is done through increasing the quantization of *ipd*+*s*/2 by the normalized difference between *w* and modulo 2 of the quantization of *ipd*+*s*/2, so that the quantization of resulting ipd^w will have *w* as the remainder when modulus 2 is taken. The reason to quantize *ipd*+*s*/2 rather than *ipd* here is to make sure that the resulting $e(ipd, w, s)$ is no less than *ipd*. Figure 2 illustrates the embedding of watermark bit *w* by mapping ranges of unwatermarked *ipd* to the corresponding watermarked ipd^w.

The watermark bit decoding function is defined as

$$d(ipd^w, s) = q(ipd^w, s) \bmod 2 \qquad (5)$$

The correctness of watermark embedding and decoding is guaranteed by the following theorems, whose proofs are in the

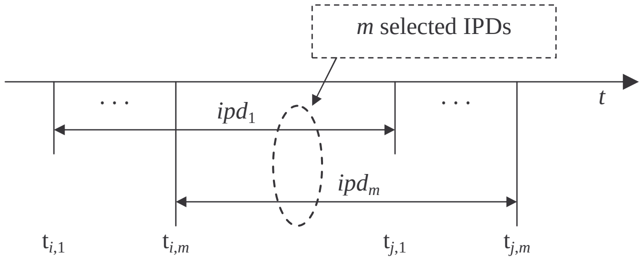

Figure 3 Embedding/Decoding Watermark Bit over the Average of Multiple (*m*) IPDs

appendix.

THEOREM 1. *For any ipd>0, s>0 and binary bit w, d(e(ipd, w, s), s) = w*.

THEOREM 2. *For any ipd>0, s>0 and binary bit w*, $0 \le e(ipd, w, s)-ipd < 2s$.

4.2 Maximum Tolerable Perturbation

Given any *ipd*>0, *s*>0, we define the *maximum tolerable perturbation* $\Delta_{\max}$ of *d*(*ipd*, *s*) as the upper bound of the perturbation over *ipd* such that

$$\forall x>0\ (x<\Delta_{\max} \Rightarrow d(ipd \pm x, s) = d(ipd, s))$$

and either

$$(d(ipd+\Delta_{\max}, s) \neq d(ipd, s)$$

or

$$d(ipd-\Delta_{\max}, s) \neq d(ipd, s))$$

That is, any perturbation smaller than $\Delta_{\max}$ on *ipd* will not change *d*(*ipd*, *s*), while a perturbation of $\Delta_{\max}$ or greater on *ipd* may change *d*(*ipd*, *s*).

We define the *tolerable perturbation range* as the subset of the perturbation range [-*D*, *D*] within which any perturbation on *ipd* is guaranteed not to change *d*(*ipd*, *s*), and the *vulnerable perturbation range* as the perturbation range outside the tolerable perturbation range.

Given any *ipd*>0, *s*>0 and binary watermark bit *w*, by definition of quantization *q* in (3) and watermark decoding function *d* in (5), it is easy to see that when $x \in (-s/2, s/2]$

$$d(e(ipd, w, s)+x, s) = d(e(ipd, w, s), s)$$

and

$$d(e(ipd, w, s)-s/2, s) \neq d(e(ipd, w, s), s).$$

This indicates that the maximum tolerable perturbation, the tolerable perturbation range and the vulnerable perturbation range of *d*(*e*(*ipd*, *w*, *s*), *s*) are *s*/2, (-*s*/2, *s*/2] and (-D, -*s*/2]$\cup$(*s*/2, D), respectively.

In summary, if the perturbation of an IPD is within the tolerable perturbation range (-*s*/2, *s*/2], the embedded watermark bit is guaranteed to be not changed by the timing attack. If the perturbation of the IPD is outside this range, the embedded watermark bit may be altered by the attacker. Therefore the larger the value of *s* (equivalently, the larger the tolerable perturbation range), the more robust the embedded watermark bit will be. However, a larger value of *s* may disturb the timing of the watermarked flow more, as the watermark bit embedding itself may add up to 2*s* delay to selected packets.

It is desirable to have a watermark embedding scheme that 1) disturbs the timing of watermarked flows as little as possible, so that the watermark embedding is less noticeable; and 2) ensures the embedded watermark bit is robust, with high probability, against timing perturbations that are outside the tolerable perturbation range (-*s*/2, *s*/2].

In the following section, we address the case when the maximum delay *D*>0 added by the attacker is bigger than the maximum tolerable perturbation *s*/2. By utilizing redundancy techniques, we develop a framework that could make the embedded watermark bit robust, with arbitrarily high probability, against arbitrarily large (and yet bounded) *iid* random timing perturbation by the attacker, as long as the flow to be watermarked contains enough packets.

5. PROBABILISTICALLY ROBUST WATERMARKING OVER IPDS

5.1 Embedding A Single Watermark Bit over the Average of Multiple IPDs

To make the embedded watermark bit probabilistically robust against larger random delays than s/2, the key is to contain and minimize the impact of the random delays on the watermark-bearing IPDs so that the impact of the random delays will fall, with high probability, within the tolerable perturbation range (-*s*/2, *s*/2].

We exploit the assumptions that: a) the attacker does not know the exact IPD(s) where the watermark bit(s) will be embedded; and, b) the random delays added by the attacker are independent and identically distributed (*iid*).

We apply the following strategies to contain and minimize the impact of random delays over the watermark-bearing IPDs:

1) Distributing watermark-bearing IPDs over a longer duration of the flow

2) Embedding a watermark bit in the average of multiple IPDs

The rationale behind these strategies is as follows. While the attacker may add a large delay to a single IPD, it is impossible to add large delays to all IPDs. In fact, random delays tend to increase some IPDs and decrease others. Therefore the impact on the average of multiple IPDs is more likely to be within the tolerable perturbation range (-*s*/2, *s*/2], even when the perturbation range [-*D*, *D*] is much larger than (-*s*/2, *s*/2].

Instead of embedding a watermark bit in one IPD, we propose to use *m*≥1 IPDs. The watermark bit is embedded in the average of the *m* IPDs (as shown in Figure 3). Since one bit is embedded in *m* IPDs, we call *m* the *redundancy number*.

Let <P$_{i,k}$, P$_{j,k}$> be the *k*-th pair (out of *m*≥1 pairs) of the packets selected to embed the watermark bit, whose timestamps are t$_{i,k}$ and t$_{j,k}$ respectively. Then we have *m* IPDs: ipd_k= t$_{j,k}$- t$_{i,k}$ (*k*=1, …, *m*). We represent the average of these *m* IPDs as

$$ipd_{avg} = \frac{1}{m} \sum_{k=1}^{m} ipd_k \qquad (6)$$

Given a desired ipd_{avg}>0, and the values for *s* and *w*, we can embed *w* into ipd_{avg} by applying the embedding function defined in (4) to ipd_{avg}. Specifically, the timing of the packets $P_{j,k}$ (*k*=1…*m*) is modified so that ipd_{avg} is adjusted by Δ, as defined in (4). To decode the watermark bit, we first collect the *m* IPDs (denoted as ipd_k^w, *k*=1…*m*) from the same *m* pairs of chosen packets and compute the average ipd_{avg}^w of $ipd_1^w \ldots ipd_m^w$. Then we can apply the decoding function defined in (5) to ipd_{avg}^{w} to decode the watermark bit.

5.2 Embedding Multiple-Bit Watermarks

We have described how to use $m \geq 1$ IPDs to embed one watermark bit with the desired robustness. Embedding this bit requires the selection of 2*m* packets, and the delay of *m* packets.

An *l*-bit watermark can be embedded simply by applying the above method *l* times, to *l* sequences of *m* packet pairs each. This is illustrated in Figure 4. It is possible to reduce the number of packets selected to (l+1)×m by making the second packet of the k^{th} (k=1,…m) packet pair chosen for embedding bit *i* the same as the first packet of the k^{th} packet pair chosen for embedding bit *i*+1.

The following information about watermark embedding is shared between the watermark embedder and the decoder. This information is assumed to be unknown to the attacker.

1) The random selection of the (l+1)×m packets and random pairing of those (l+1)×m packets for embedding and decoding the watermark.
2) The redundancy number *m*.
3) The number of watermark bits *l*.
4) The quantization step size *s*.

5.3 Attacker's Impact over the Average of Multiple IPDs

Let $d_{i,k}$ and $d_{j,k}$ be the random variables that denote the random delays added by the attacker to packets $P_{i,k}$ and $P_{j,k}$ respectively for k=1,…,m. By assumption, $d_{i,k}$ and $d_{j,k}$ (k=1,…,m) are independent and identically distributed. Therefore $d_{i,1}$,…,$d_{i,m}$ and $d_{j,1}$,…,$d_{j,m}$ form two random samples from the distribution of random delays added by the attacker.

Let X_k=$d_{j,k}$-$d_{i,k}$ be the random variable that denotes the impact of these random delays on ipd_k and $\overline{X_m}$ be the random variable that denotes the overall impact of random delay on ipd_{avg}. From (6) we have

$$\overline{X_m} = \frac{1}{m}\sum_{k=1}^{m}(d_{j,k}-d_{i,k}) = \frac{1}{m}\sum_{k=1}^{m}X_k \qquad (7)$$

Therefore the impact of the random delay by the attacker over ipd_{avg} equals the sample mean of $X_1 \ldots X_m$.

We define the probability that the impact of the timing perturbation by the attacker is within the tolerable perturbation range (-*s*/2, *s*/2] as the *watermark bit robustness p*, which can be expressed as $p = \Pr(\ |\overline{X_m}| < s/2\ )$.

Similarly we define the probability that the impact of the timing perturbation by the attacker is out of the tolerable perturbation range (-*s*/2, *s*/2] as the *watermark bit vulnerability*, which can be quantitatively expressed as $\Pr(\ |\overline{X_m}| \geq s/2\ )$.

Let σ^2 be the variance of the random delay added by the attacker. Because the maximum delay that may be added by the attacker is assumed to be bounded, σ^2 is finite.

From the properties of the mean and variance of random variables, we have $E(X_k) = E(d_{j,k}) - E(d_{i,k}) = 0$ and $Var(X_k) = Var(d_{j,k}) + Var(d_{i,k}) = 2\sigma^2$. We further have $E(\overline{X_m}) = 0$ and $Var(\overline{X_m}) = 2\sigma^2/m$. This indicates that the probability distribution of $\overline{X_m}$ is more concentrated around its mean than X_k.

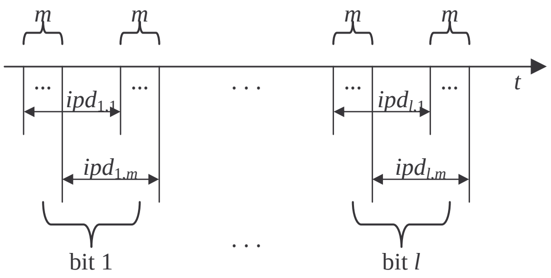

Figure 4. Embedding *l*-bit watermark into *l* sequences of IPDs

According to the Chebyshev inequality in statistics[4], for any random variable *X* with finite variance *Var*(*X*) and for any *t*>0, $\Pr(|\ X - E(X)\ | \geq t) \leq Var(X)/t^2$. This means that the probability that a random variable deviates from its mean by more than *t* is bounded by $Var(X)/t^2$.

By applying the Chebyshev inequality to $\overline{X_m}$ with *t*=*s*/2, we have

$$\Pr(|\overline{X_m}| \geq s/2) \leq 8\sigma^2/ms^2 \qquad (8)$$

This means that the probability that the overall impact of *iid* random delays on ipd_{avg} is outside the tolerable perturbation range (-s/2, s/2] is bounded. In addition, that probability can be reduced to be arbitrarily close to 0 by increasing *m*, the number of redundant IPDs averaged for embedding the watermark. This result holds true regardless of the mean or the variance of the *iid* random delays added by the attacker, or of the maximum quantization delay allowed for watermark embedding.

6. ANALYSIS ON THE DISTRIBUTION OF WATERMARK BIT ROBUSTNESS

In the previous section, we established an upper bound for watermark bit vulnerability $\Pr(|\overline{X_m}| \geq s/2)$ through the Chebyshev inequality. We now show how to apply the well-known Central Limit Theorem of statistics[4] to get an accurate approximation to the distribution of the robustness of the embedded watermark bit.

Central Limit Theorem. *If the random variables X_1, …, X_n form a random sample of size n from a given distribution X with mean μ and finite variance σ^2,then for any fixed number x*

$$\lim_{n\to\infty}\Pr[\frac{\sqrt{n}(\overline{X_n}-\mu)}{\sigma} \leq x] = \Phi(x) \qquad (9)$$

where $\Phi(x) = \int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}e^{-\frac{u^2}{2}}du$.

The theorem indicates that whenever a random sample of size *n* is taken from any distribution with mean μ and finite variance σ^2, the sample mean $\overline{X_n}$ will be approximately normally distributed with mean μ and variance σ^2/n, or equivalently the distribution of random variable $\sqrt{n}(\overline{X_n}-\mu)/\sigma$ will be approximately a standard normal distribution.

Let σ^2 denote the variance of the distribution of the random delays added by the attacker (*i.e.,* let $Var(d_{i,k}) = Var(d_{j,k}) = \sigma^2$).

Applying the Central Limit Theorem to random sample $X_1 = d_{j,1}\text{-}d_{i,1}, \ldots, X_m = d_{j,m}\text{-}d_{i,m}$, where $Var(X_k) = Var(d_{i,k})+Var(d_{j,k}) = 2\sigma^2$ and $E(X_k) = E(d_{j,k})\text{-}E(d_{i,k}) = 0$, we have

$$\begin{aligned} &\Pr[\frac{\sqrt{m}(\overline{X}_m - E(X_i))}{\sqrt{Var(X_i)}} < x] \\ &= \Pr[\frac{\sqrt{m}\,\overline{X}_m}{\sqrt{2}\sigma} < x] \\ &\approx \Phi(x) \end{aligned} \tag{10}$$

or

$$\Pr[|\frac{\sqrt{m}\,\overline{X}_m}{\sqrt{2}\sigma}| < x] \approx 2\Phi(x) - 1 \tag{11}$$

Therefore,

$$\begin{aligned} p &= \Pr[|\overline{X}_m| < \frac{s}{2}] \\ &= \Pr[|\frac{\sqrt{m}\,\overline{X}_m}{\sqrt{2}\sigma}| < \frac{s\sqrt{m}}{2\sqrt{2}\sigma}] \\ &\approx 2\Phi(\frac{s\sqrt{m}}{2\sqrt{2}\sigma}) - 1 \end{aligned} \tag{12}$$

This means that the distribution of the watermark bit robustness is approximately normally distributed with zero mean and variance $2\sigma^2/m$.

Equation (12) confirms the result of equation (8). Figure 5 illustrates how the distribution of the impact of random timing perturbation by the attacker can be "squeezed" into the tolerable perturbation range by increasing the number of redundant IPDs averaged.

Equation (12) also gives us an accurate estimate of the watermark bit robustness. For example, assume the maximum delay by the attacker is normalized to be 1 time unit, the random delays added by the attacker are uniformly distributed over [0, 1] (whose variance σ^2 is 1/12), s=0.4 units and m=12, then $\Pr[|\overline{X}_{12}| < 0.2] \approx 2\Phi(1.2 \times \sqrt{2}) - 1 \approx 91\%$. We can expect the impact of the random delays on the average of those 12 IPDs, with about 91% probability, will fall within the range [-0.2, 0.2]. Table 1 shows the estimation and simulation results of watermark bit robustness with uniformly distributed random delays over [0, 1], s=0.4 and various sample values for m. It demonstrates that the Central Limit Theorem can give us a precise estimate with a sample size as small as m=7.

Table 1 Watermark Bit Robustness Estimation & Simulation with Uniformly Distributed Random Delay over [0, 1], *s*=0.4

m	**7**	**8**	**9**	**10**	**11**	**12**
Estimated Robustness (%)	80.46	83.32	85.84	87.86	89.58	91.02
Simulated Robustness (%)	80.27	83.27	85.68	87.79	89.54	91.02

From equation (12), we can also see that it is easier to achieve the same robustness by increasing s than by increasing m. For example, if s were reduced by a factor of 2, m would have to be increased by a factor of 4 to maintain the same robustness level.

7. WATERMARK DETECTION

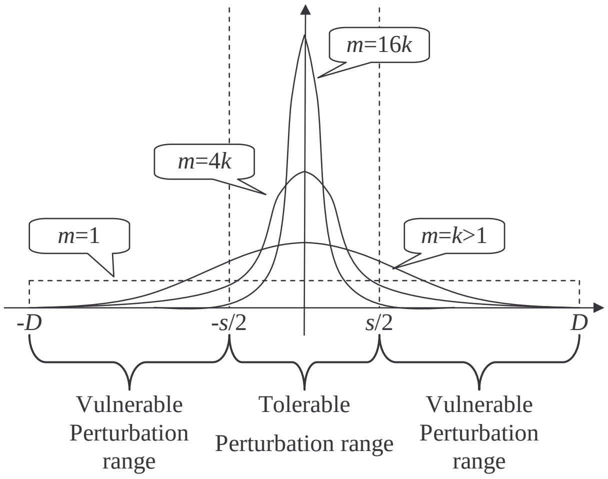
m=16k
m=4k
m=1
m=k>1
-D
-s/2
s/2
D
Vulnerable Perturbation range
Tolerable Perturbation range
Vulnerable Perturbation range

Figure 5. Probability Distribution of the Impact of Random Delays over the Average of Multiple (*m*) IPDs

Watermark detection refers to the process of determining if a given watermark is embedded in the IPDs of a specific connection or flow.

Let the information shared between the watermark embedder and decoder be represented as <S, m, l, s, wm>, where $S()$ is the selection function that returns $(l+1)\times m$ packets, $m \geq 1$ is the number of redundant pairs of packets in which to embed one watermark bit, $l>0$ is the length of the watermark in bits, $s>0$ is the quantization step size, and wm is the l-bit watermark to be detected. Let f denote the flow to be examined and wm_f denote the decoded l bits from flow f.

The watermark detector works as follows:

1) Decode the l-bit wm_f from flow f.
2) Compare the decoded wm_f with wm.
3) Report that watermark wm is detected in flow f if the Hamming distance between wm_f and wm, represented as H(wm_f, wm), is less than or equal to h, where h is a threshold parameter determined by the user, and $0 \leq h < l$.

The rationale behind using the Hamming distance rather than requiring an exact match to detect the presence of wm is to increase the robustness of the watermark detector against countermeasures by the attacker. Given any quantization step size s, there is always a slight chance that the embedded watermark bit is corrupted by countermeasures by the attacker no matter how many redundant pairs of packets are used. Let $0<p<1$ be the probability that each embedded watermark bit will survive the timing perturbation by the attacker. Then the probability that all l bits survive the timing perturbation by the attacker will be p^l. When l is reasonably large, p^l will tend to be small unless p is very close to 1.

By using the Hamming distance h to detect watermark wm_f, the expected watermark detection rate will be

$$\sum_{i=0}^{h} \binom{l}{i} p^{l-i}(1-p)^i \tag{13}$$

For example, for the values p=0.9102, l=24, h=5, the expected watermark detection rate with exact bit match would be p^l =10.45%. For the same values of p, l, and h, the expected watermark detection rate using a Hamming distance h=5 would be

98.29%.

It is possible for the watermark detector to mistakenly report a watermark for a flow in which no watermark has been embedded. It is termed a *collision* between *wm* and *f* if $H(wm_f, wm) \le h$ for an unwatermarked flow *f*.

Assuming the *l*-bit wm_f extracted from random flow *f* is uniformly distributed, then the expected watermark collision probability between any particular watermark *wm* and a random flow *f* will be

$$\sum_{i=0}^{h}\binom{l}{i}\left(\frac{1}{2}\right)^{l} \qquad (14)$$

Figure 6 shows the derived probability distribution of the expected watermark detection and collision rates with l=24 and p=0.9102. Given any watermark bit number l>1 and any watermark bit robustness $0<p<1$, the larger the Hamming distance threshold *h* is, the higher the expected detection rate will be. However, a larger Hamming distance threshold tends to increase the collision (false positive) rate of the watermark detection at the same time. An optimal Hamming distance threshold would be one that gives a high expected detection rate, while keeping the false positive rate low.

Given any quantization step size s>0, any desired watermark collision probability P_c>0, and any desired watermark detection rate $0<P_d<1$, we can determine the appropriate Hamming distance threshold $0<h<l$. Assuming that *h* is chosen such that $h < l/2$, then we have

$$\sum_{i=0}^{h}\binom{l}{i}\left(\frac{1}{2}\right)^{l} \le \sum_{i=0}^{h}\binom{l}{h}\left(\frac{1}{2}\right)^{l} \le (h+1)\frac{l^h}{2^l} \qquad (15)$$

Because $\lim_{l\to\infty}\frac{l^h}{2^l}=0$, we can always make the expected watermark collision probability $\sum_{i=0}^{h}\binom{l}{i}\left(\frac{1}{2}\right)^{l} < P_c$ by having sufficiently large watermark bit number *l*. Since $\sum_{i=0}^{h}\binom{l}{i}p^{l-i}(1-p)^{i} \ge p^{l}$, we can always make the expected detection rate $\sum_{i=0}^{h}\binom{l}{i}p^{l-i}(1-p)^{i} > p_d$ by having $0<p<1$ sufficiently close to 1. From inequality (8), this can be accomplished by increasing the redundancy number *m* regardless of the value of *s* andσ.

Therefore, in theory, our watermark based correlation scheme can, with arbitrarily small averaged adjustment of inter-packet timing (for embedding the watermark), achieve arbitrarily close to a 100% watermark detection rate and arbitrarily close to a 0% watermark collision probability at the same time against arbitrarily large (but bounded) independent and identically distributed (iid) random timing perturbation of arbitrary distribution, as long as there are enough packets in the flow to be watermarked.

7.1 Limitation

In theory, our watermark correlation is effective and robust against random delays that are independent and identically distributed (*iid*) over the set of watermarked packets. For random delays that are independent but have different distributions over the set of watermarked packets, the maximum tolerable perturbation *s*/2 may have to be greater than a specific non-zero value to achieve an arbitrarily high watermark detection rate and arbitrarily low watermark collision rate at the same time. This is due to the fact that the random variable $X_k = d_{j,k}-d_{i,k}$ may have a non-zero mean if $d_{j,k}$ and $d_{i,k}$ are of different distributions. In addition, our watermark correlation method is not as robust against non-independent random delays. An extreme case would be when the attacker knows exactly which packets have been delayed and by how much, making it much easier to corrupt the embedded watermark bits.

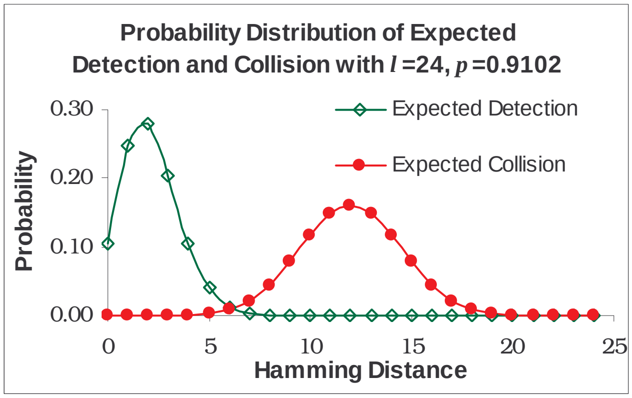

Figure 6. Distribution of Expected Watermark Detection and Collision

8. EXPERIMENTS

The goal of the experiments is to answer the following questions about watermark-based correlation (as well as existing timing-based correlation) in the face of random timing perturbation by the attacker:

1) How vulnerable are existing (passive) timing-based correlation schemes to random timing perturbations?
2) How robust is watermark-based correlation against random timing perturbations?
3) How effective is watermark-based correlation in correlating the encrypted flows that are perturbed in timing?
4) What is the collision (false positive) rate of watermark-based correlation?
5) How well do the models of watermark bit robustness, watermark detection rate and watermark collision rate predict the measured values?

We have used two flow sets, labeled FS1 and FS2 in our experiments. FS1 is derived from over 49 million packet headers of the Bell Labs-1 Traces of NLANR[9]. It contains 121 SSH flows that have at least 600 packets and that are at least 300 seconds long. FS2 contains 1000 telnet flows generated from an empirically-derived distribution[3] of telnet packet inter-arrival times, using the tcplib[2] tool.

8.1 Correlation True Positive Experiment

To answer the first three questions, we have conducted the following experiment. First, we used an existing, *passive* timing-based correlation method called IPD-Based Correlation[16] to correlate each flow in FS1 with the same flow, after the interpacket delays of the flow have been randomly perturbed. If the flow and the perturbed flow are reported correlated, it is

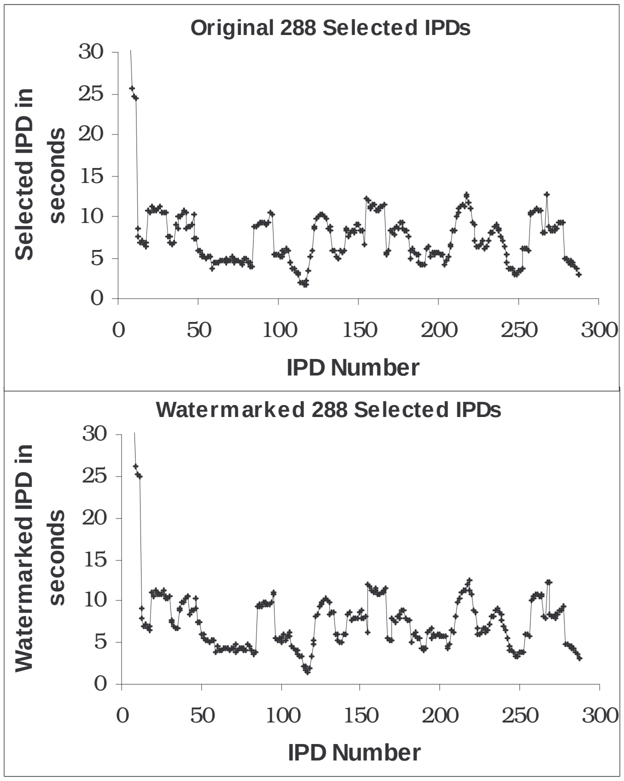

Figure 7 Comparison of 288 Selected IPDs before and after Watermark Embedding

considered a *true positive* (TP) of the correlation in the presence of timing perturbation. Second, we embedded a random 24-bit watermark into each flow of FS1 and FS2, with redundancy number *m*=12, and quantization step size *s*=400ms for each watermark bit. The embedding of the 24-bit watermark required 300 packets to be selected; 288 packets were delayed to embed the watermark. Figure 7 shows the effect of the watermark embedding, and illustrates that the embedding is far from being obvious. Third, we randomly perturbed the packet timing of the watermarked flows of FS1 and FS2. It is considered a *true positive* of watermark-based correlation if the embedded watermark can be detected from the timing perturbed watermarked flows, with a Hamming distance threshold *h*=5. Finally, we calculated the expected detection rate from equations (12) and (13) under various maximum delays of the random timing perturbation.

Each data point in Figure 8 shows the average of 100 separate experiments measuring the true positive rates of IPD-based Correlation and watermark-based correlation on FS1 and FS2. The results clearly indicate that IPD-based correlation is vulnerable to even moderate random timing perturbation. Without timing perturbation, IPD-based correlation is able to successfully correlate 93.4% of the SSH flows of FS1. However, with a maximum 100ms random timing perturbation, the true positive rate of IPD-based correlation drops to 45.5%, and for a 200ms maximum delay, the rate drops to 21.5%.

In contrast, the proposed watermark-based correlation of the flows in FS1 and FS2 is able to achieve virtually a 100% true positive rate, up to a maximum 600ms random timing perturbation. With a maximum 1000ms timing perturbation, the true positive rates of watermark-based correlation for FS1 and FS2 are 84.2% and 97.32%, respectively. It can be seen that the measured watermark-based correlation true positive rates are well approximated by the estimated values, based on the watermark detection rate model (equation (13)). In particular, the true positive rate measurements of FS2 are almost identical to the estimated values at all perturbation levels.

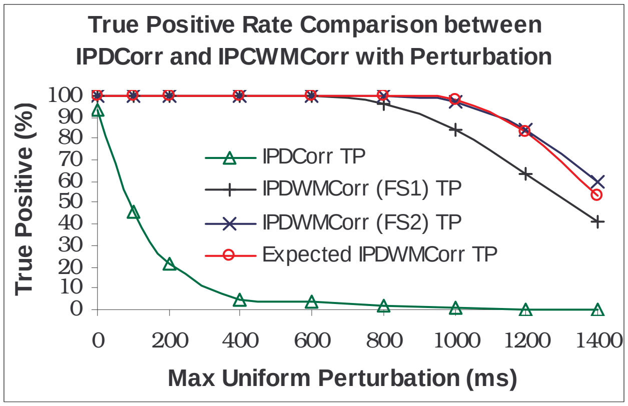

Figure 8. Correlation True Positive Rates under Random Timing Perturbs

8.2 Correlation False Positive Experiment

As explained above, there is a non-zero probability that an un-watermarked flow will happen to exhibit the randomly chosen watermark. This case is considered a correlation collision, or false positive. According to our correlation collision model (14), the collision rate is determined by the number of watermark bits *l* and the Hamming distance threshold *h*.

We therefore experimentally investigated the following, for varying values of the Hamming distance threshold *h*:

1) Collision rates between a given flow and 10,000~1,000,000 randomly generated 24-bit watermarks

2) Collision rates between a given 24-bit watermark and 10,000~1,000,000 randomly generated (using tcplib) telnet flows.

Figure 9 shows the results. For each data point in Figure 9, 100 experiments were run, and the average is shown.

The measured collision rates and expected values are very close, validating our model. In addition, the results show that the collision rate can be controlled to a low value by appropriate selection of the Hamming distance threshold.

8.3 Tradeoff between Watermark Detection Rate and Redundancy Number

Equation (12) gives us the quantitative tradeoff between the expected watermark bit robustness and redundancy number *m*. With a given watermark bit robustness *p*, equation (13) gives us the expected watermark detection rate.

To verify the validity and accuracy of our models of watermark bit robustness and watermark detection rate, we embedded a random 24-bit watermark into each flow in FS1 and FS2, for

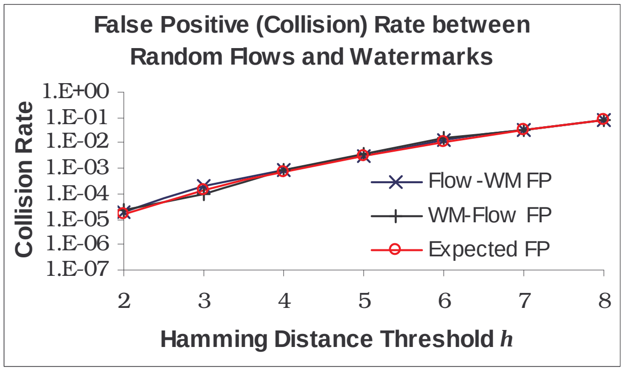

Figure 9. Correlation False Positive (Collision) Rate vs Hamming Distance Threshold *h*

different redundancy numbers *m*=7,8,9,10,11,12. The quantization step *s* was set to 400ms for each watermark bit. Then we perturbed the watermarked flows with 1000ms maximum random delays. Finally, we measured the watermark detection rate of the perturbed, watermarked flows.

Figure 10 shows the average of 100 experiments for the measured watermark detection rates of FS1, and the average of 10 experiments for the measured watermark detection rates of FS2. Also shown is the expected detection rate derived from equations (12) and (13) for the various values of the redundancy number *m*. The detection rates of FS2 are very close to the expected values, while the detection rates of FS1 are similar to but lower than the expected values. These results validate our models of watermark bit robustness and watermark detection rate.

9. CONCLUSIONS AND FUTURE WORK

Tracing attackers' traffic through stepping stones is a challenging problem, since they have a variety of countermeasures at their disposal to evade correlation of connections across stepping stones. In particular, random timing perturbation by the attacker greatly reduces the effectiveness of passive, timing-based correlation techniques.

We presented an active timing-based approach to deal with random timing perturbation. By embedding a watermark into the packet timing, with sufficient redundancy we can correlate in a way that is probabilistically robust against random timing perturbations. Our experiments show that watermark-based correlation is substantially more effective than passive, timing-based correlation in the presence of random timing perturbations.

For independent and identically distributed (*iid*) random delays added by the attacker, our model reveals a rather surprising theoretical result on the limits of watermark-based correlation: *the proposed watermark based correlation scheme can, with arbitrarily small average adjustment of inter-packet timing, achieve arbitrarily close to 100% watermark detection (true positive) rate and arbitrarily close to 0% collision (false positive) probability at the same time against arbitrarily large (but bounded) independent and identically distributed (iid) random timing perturbations of arbitrary distribution, as long as there are enough packets in the flow to be watermarked.*

We also developed models of the tradeoff between the watermark detection (or true positive) rate and watermark collision (or false positive) rate. Our experimental results validate the accuracy of these tradeoff models. Thus our tradeoff models are of practical value in optimizing the overall effectiveness of watermark-based correlation in real world situations.

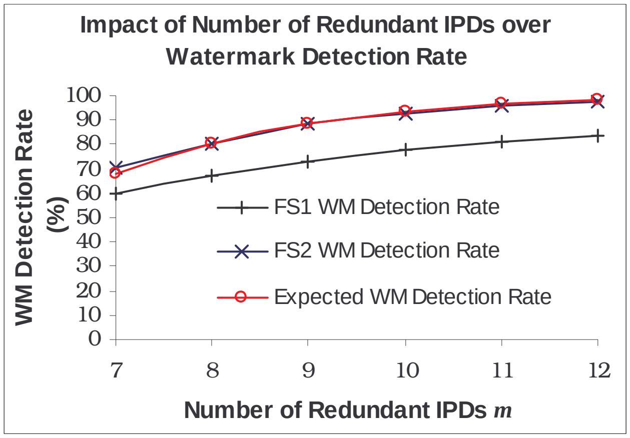

Figure 10. Watermark Detection Rates vs Redundancy Number *m*

Future research work includes how to effectively correlate connections when the attacker 1) reorders the packets; 2) drops/retransmits some packets; or, 3) adds padding packets ("chaff" [5]).

10. REFERENCES

[1] I. J. Cox, M. L. Miller and J. A. Bloom. *Digital Watermarking*. Morgan-Kaufmann Publishers, 2002.

[2] P. B. Danzig and S. Jamin. tcplib: A Library of TCP Internetwork Traffic Characteristics. USC Technical Report, USC-CS-91-495.

[3] P. B. Danzig, S. Jamin, R. Cacerest, D. J. Mitzel and E. Estrin. An Empirical Workload Model for Driving Wide-Area TCP/IP Network Simulations. In *Journal of Internetworking* 3:1, pages 1–26 March 1992.

[4] M. H. DeGroot. *Probability and Statistics*. Addison-Wesley Publishing Company, 1989.

[5] D. Donoho, A.G. Flesia, U. Shanka, V. Paxson, J. Coit and S. Staniford. Multiscale Stepping Stone Detection: Detecting Pairs of Jittered Interactive Streams by Exploiting Maximum Tolerable Delay. In *Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002)*, October, 2002. Springer Verlag Lecture Notes in Computer Science, #2516.

[6] M. T. Goodrich. Efficient Packet Marking for Large-Scale IP Traceback. In *Proceedings of 9th ACM Conference on Computer and Communication Security CCS'02*, pages 117−126, October 2002.

[7] H. Jung, et al. Caller Identification System in the Internet Environment. In *Proceedings of 4th USENIX Security Symposium*, 1993.

[8] S. Kent, R. Atkinson. Security Architecture for the Internet Protocol. *IETF RFC 2401*, September 1998.

[9] NLANR Trace Archive. http://pma.nlanr.net/Traces/long/.

[10] OpenSSH. http://www.openssh.com.

[11] S. Savage, D. Wetherall, A. Karlin and T. Anderson. Practical Network Support for IP Traceback. In *Proceedings of the ACM SIGCOMM 2000*, April 2000.

[12] S. Snapp, et al. DIDS (Distributed Intrusion Detection System) – Motivation, Architecture and Early Prototype. In *Proceedings of 14th National Computer Security Conference*, pages 167–176, 1991.

[13] D. Song and A. Perrig. Advanced and Authenticated Marking Scheme for IP Traceback. In *Proceedings of IEEE INFOCOM'01*, April 2001.

[14] S. Staniford-Chen, L. T. Heberlein. Holding Intruders Accountable on the Internet. In *Proceedings of* the *IEEE Symposium on Security and Privacy*, May 1995.

[15] C. Stoll. *The Cuckoo's Egg: Tracking Spy through the Maze of Computer Espionage*. Pocket Books, October 2000.

[16] X. Wang, D. S. Reeves and S.F. Wu. Inter-Packet Delay-Based Correlation for Tracing Encrypted Connections through Stepping Stones. In *D. Gollmann, G. Karjoth and M. Waidner, editors, 7th European Symposium on Research in Computer Security – ESORICS 2002,* October 2002. Springer-Verlag Lecture Notes in Computer Science #2502.

[17] X. Wang, D. S. Reeves, S. F. Wu and J. Yuill. Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework. In *Proceedings of 16th International Conference on Information Security (IFIP/Sec'01)*, June, 2001.

[18] T. Ylonen, et al. SSH Protocol Architecture. *IETF Internet Draft: draft-ietf-secsh-architecture-4.txt,* July 2003.

[19] K. Yoda and H. Etoh. Finding a Connection Chain for Tracing Intruders. In F. Guppens, Y. Deswarte, D. Gollmann and M. Waidner, editors, *6th European Symposium on Research in Computer Security – ESORICS 2000,* October 2000. Springer-Verlag Lecture Notes in Computer Science #1895

[20] Y. Zhang and V. Paxson. Detecting Stepping Stones. In *Proceedings of the 9th USENIX Security Symposium*, pages 171–184, 2000.

11. APPENDIX

Proof of Theorem 1

Given any $ipd>0$, we can find unique $a\geq 0$ and $-s/2<b\leq s/2$ such that $ipd+s/2=a\times s+b$. Then we have $q((ipd+s/2), s)=a$ and $e(ipd, w, s)=[a+((w-a) \bmod 2 +2) \bmod 2]\times s$. Therefore

$$\begin{aligned}
&d(e(ipd, w, s), s)\\
&= q(e(ipd, w, s), s) \bmod 2\\
&= q([a+((w-a) \bmod 2 +2) \bmod 2]\times s) \bmod 2\\
&= round(a+((w-a) \bmod 2 +2) \bmod 2) \bmod 2\\
&= (a+((w-a) \bmod 2 +2) \bmod 2) \bmod 2\\
&= (a+w-a+2) \bmod 2\\
&= w
\end{aligned}$$

Proof of Theorem 2

Given any $ipd>0$ and $s>0$, assume $round(ipd/s+1/2)=i$, by definition of $round(x)$, we have $ipd/s+1/2 \in (i-1/2, i+1/2]$. That is $i-1<ipd/s\leq i$ or $(i-1)\times s<ipd\leq i\times s$. Replace i with $round(ipd/s+1/2)$, we have $round(ipd/s+1/2)\times s-s<ipd\leq round(ipd/s+1/2)\times s$.

By (4) we have

$$\begin{aligned}
&e(ipd, w, s), s)\\
&= [q((ipd+s/2), s)+(w-(q((ipd+s/2), s) \bmod 2+2) \bmod 2]\times s\\
&\geq q((ipd+s/2), s)\times s\\
&= round(ipd/s+1/2)\times s\\
&\geq ipd
\end{aligned}$$

and

$$\begin{aligned}
&e(ipd, w, s), s)\\
&= [q((ipd+s/2), s)+(w-(q((ipd+ s/2), s) \bmod 2+2) \bmod 2]\times s\\
&\leq [q((ipd+s/2), s)+1]\times s\\
&= round(ipd/s+1/2)\times s+s\\
&< ipd+2s
\end{aligned}$$

Therefore, $0\leq e(ipd, w, s)-ipd<2s$.