

# Robust Correlation of Encrypted Attack Traffic Through Stepping Stones By Watermarking The Interpacket Timing

Xinyuan Wang  
Dept. of Computer Science  
North Carolina State University  
Raleigh, NC 27695  
[xwang5@unity.ncsu.edu](mailto:xwang5@unity.ncsu.edu)  
ACM No. 3543246

Douglas S. Reeves (Advisor)  
Dept. of Computer Science  
North Carolina State University  
Raleigh, NC 27695  
[reeves@csc.ncsu.edu](mailto:reeves@csc.ncsu.edu)

## 1 Problem and Motivation

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<sup>[15]</sup>. To identify attackers behind stepping stones, it is necessary to be able to correlate connections through stepping stones, even if those connections are encrypted and perturbed by the intruder to avoid traceability.

The timing-based approach is currently the most capable and promising method for correlating encrypted connections. However, previous timing-based approaches are vulnerable to packet timing perturbations introduced by the attacker at stepping stones. In particular, the attacker can 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 redundancy techniques, we have developed a robust correlation scheme that reveals a rather surprising result on the inherent limits of independent and identically distributed (*iid*) random timing perturbations over sufficiently long flows. *Our active watermarking correlation scheme can achieve, at least in theory, a detection (true positive) rate arbitrarily close to 100%, and a watermark collision (false positive) rate arbitrarily close to 0 at the same time, for an arbitrarily large (but bounded) independent and identically distributed (iid) random timing perturbation of arbitrary distribution, with arbitrarily small average adjustment of inter-packet timing, as long as there are enough packets in the flow to be watermarked.* We also identify the tradeoffs between the defining characteristics of timing perturbation and the 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.

## 2 Background and 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<sup>[7]</sup> and DIDS<sup>[12]</sup>) collects and tracks user login activities at each stepping stone. Because the attackers can modify, delete, or forge local user login information, host activity based correlation could easily be defeated.

Approaches based on connection content (e.g., Thumbprinting<sup>[14]</sup> and SWT<sup>[17]</sup>) 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<sup>[16]</sup>, Deviation-based<sup>[19]</sup> and ON/OFF-based<sup>[20]</sup>) use the arrival and/or departure times of packets to correlate connections. While the timing based approaches are shown to be quite effective in correlating encrypted connections, they are vulnerable to malicious timing perturbation by attackers.

---

This work was originally published in the Proceedings of the *10th ACM Conference on Computer and Communications Security (CCS 2003)*.

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 type of timing perturbation 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 timing 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 represents an important first step toward a better understanding of the inherent limitations of timing perturbation by the attacker on timing-based correlation. The important theoretical result is that correlation is still achievable for sufficiently long flows despite certain type of timing perturbations. What left open are the question whether correlation is achievable for arbitrarily distributed (rather than Pareto distribution conserving) random timing perturbation, and an analysis of the achievable tradeoff of the false positive and true positive rates.

In the following sections we show, with our active watermarking correlation, that for sufficient long flows, it is indeed possible to achieve both high correlation true positive rate and low correlation false positive rate at the same time against arbitrarily large *iid* random timing perturbations of arbitrary distribution.

### 3 Approach and Uniqueness

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

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

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.

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

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

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

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

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  $P_i$  ( $i = 1, \dots, n$ ), 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.

#### 3.2 Basic Watermark 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

$$q(ipd, s) = \text{round}(ipd / s) \quad (3)$$

where  $\text{round}(x)$  is the function that rounds off real number  $x$  to its nearest integer (i.e.,  $\text{round}(x) = i$  for any  $x \in (i - 1/2, i + 1/2]$ ). 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

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

where  $\Delta = (w - (q(ipd + s/2, s) \bmod 2) + 2) \bmod 2$ . Figure 1 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

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

### 3.2.1 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.

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  $s/2$  (or equivalently, the perturbation is outside the tolerable perturbation range  $[-D, D]$ ). 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.

### 3.3 Probabilistically Robust Watermarking Over 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

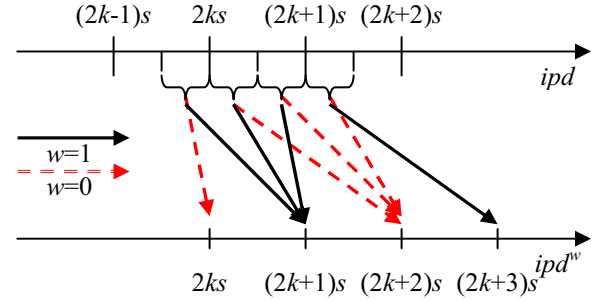


Figure 1. Mapping between Unwatermarked  $ipd$  and Watermarked  $ipd^w$  to Embed Watermark Bit

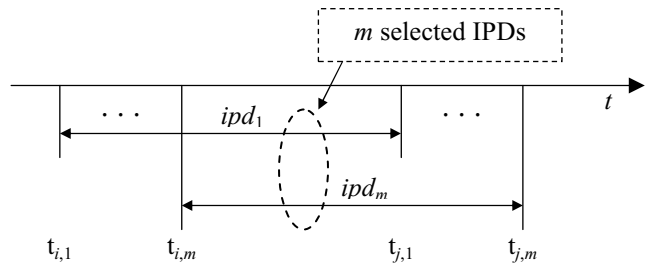


Figure 2 Embedding/Decoding Watermark Bit over the Average of Multiple ( $m$ ) IPDs

attacker are independent and identically distributed (*iid*).

Instead of embedding a watermark bit in one IPD, we propose to use  $m \geq 1$  IPDs. The watermark bit is embedded in the average of the  $m$  IPDs (as shown in Figure 2). Since one bit is embedded in  $m$  IPDs, we call  $m$  the *redundancy number*.

Let  $\langle P_{i,k}, P_{j,k} \rangle$  be the  $k$ -th pair (out of  $m \geq 1$  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:  $ipd_k = t_{j,k} - t_{i,k}$  ( $k=1, \dots, m$ ). We represent the average of these  $m$  IPDs as

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

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  $P_{j,k}$  ( $k=1 \dots m$ ) is modified so that  $ipd_{avg}$  is adjusted by  $\Delta$ , as defined in (4). To decode the watermark bit, we first collect the  $m$  IPDs (denoted as  $ipd_k^w$ ,  $k=1 \dots m$ ) from the same  $m$  pairs of chosen packets and compute the average  $ipd_{avg}^w$  of  $ipd_1^w \dots ipd_m^w$ . Then we can apply the decoding function defined in (5) to  $ipd_{avg}^w$  to decode the watermark bit.

### 3.3.1 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  $k=1, \dots, m$ . By assumption,  $d_{i,k}$  and  $d_{j,k}$  ( $k=1, \dots, m$ ) are independent and identically distributed. Therefore  $d_{i,1}, \dots, d_{i,m}$  and  $d_{j,1}, \dots, d_{j,m}$  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

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

Therefore the impact of the random delay by the attacker over  $ipd_{avg}$  equals the sample mean of  $X_1 \dots X_m$ . 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  $\sigma^2$  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,  $\sigma^2$  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  $X_k$ .

According to the Chebyshev inequality in statistics<sup>[4]</sup>, 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

$$\Pr(|\overline{X}_m| \geq s/2) \leq 8\sigma^2/ms^2 \quad (8)$$

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.

### 3.3.2 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<sup>[4]</sup> to get an accurate approximation to the distribution of the robustness of the embedded watermark bit.

Central Limit Theorem. *If the random variables  $X_1, \dots, X_n$  form a random sample of size  $n$  from a given distribution*

$X$  with mean  $\mu$  and finite variance  $\sigma^2$ , then for any fixed number  $x$

$$\lim_{n \rightarrow \infty} \Pr\left[\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \leq x\right] = \Phi(x) \quad (9)$$

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  $\mu$  and finite variance  $\sigma^2$ , the sample mean  $\bar{X}_n$  will be approximately normally distributed with mean  $\mu$  and variance  $\sigma^2/n$ , or equivalently the distribution of random variable  $\sqrt{n}(\bar{X}_n - \mu)/\sigma$  will be approximately a standard normal distribution.

Let  $\sigma^2$  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  $X_1 = d_{j,1} - d_{i,1}, \dots, X_m = d_{j,m} - d_{i,m}$ , where  $Var(X_k) = Var(d_{i,k}) + Var(d_{j,k}) = 2\sigma^2$  and  $E(X_k) = E(d_{j,k}) - E(d_{i,k}) = 0$ , we have

$$\Pr\left[\frac{\sqrt{m}(\bar{X}_m - E(X_i))}{\sqrt{Var(X_i)}} < x\right] = \Pr\left[\frac{\sqrt{m}\bar{X}_m}{\sqrt{2}\sigma} < x\right] \approx \Phi(x) \quad (10)$$

or

$$\Pr\left[\left|\frac{\sqrt{m}\bar{X}_m}{\sqrt{2}\sigma}\right| < x\right] \approx 2\Phi(x) - 1 \quad (11)$$

Therefore,

$$p = \Pr\left[\left|\bar{X}_m\right| < \frac{s}{2}\right] = \Pr\left[\left|\frac{\sqrt{m}\bar{X}_m}{\sqrt{2}\sigma}\right| < \frac{s\sqrt{m}}{2\sqrt{2}\sigma}\right] \approx 2\Phi\left(\frac{s\sqrt{m}}{2\sqrt{2}\sigma}\right) - 1 \quad (12)$$

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 3 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  $\sigma^2$  is  $1/12$ ),  $s=0.4$  units and  $m=12$ , then  $\Pr\left[\left|\bar{X}_{12}\right| < 0.2\right] \approx 2\Phi(1.2 \times \sqrt{2}) - 1 \approx 91\%$ . 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  $[-0.2, 0.2]$ .

### 3.4 Watermark Detection

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

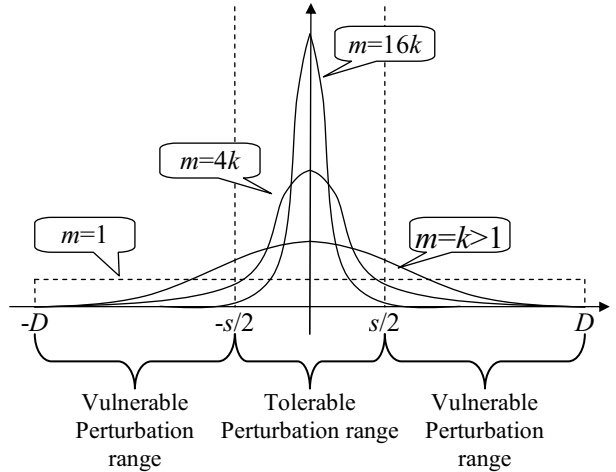
Let the information shared between the watermark embedder and decoder be represented as  $\langle S, m, l, s, wm \rangle$ , where  $S()$  is the selection function that returns  $(l+1) \times m$  packets,  $m \geq 1$  is the number of redundant pairs of packets in which to embed one watermark bit,  $l > 0$  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  $0 \leq h < l$ .

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

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



**Figure 3. Probability Distribution of the Impact of Random Delays over the Average of Multiple ( $m$ ) IPDs**

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  $H(wm_f, wm) \leq h$  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

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

Figure 4 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.

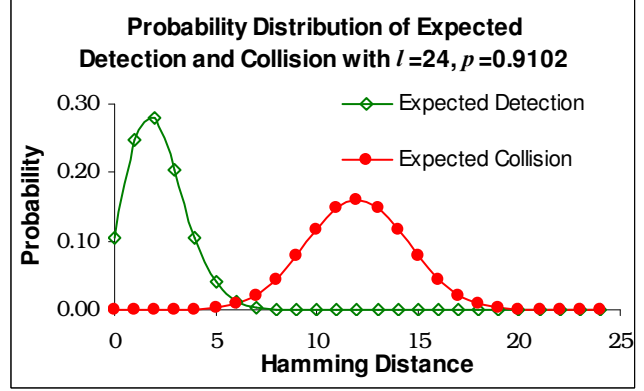


Figure 4. Distribution of Expected Watermark Detection and Collision

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

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

Because  $\lim_{l \rightarrow \infty} \frac{l^h}{2^l} = 0$ , 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  $\sum_{i=0}^h \binom{l}{i} p^{l-i} (1-p)^i \geq p^l$ , 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  $s$  and  $\sigma$ .

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.

### 3.5 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 NLNR<sup>[9]</sup>. 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<sup>[3]</sup> of telnet packet inter-arrival times, using the tcplib<sup>[2]</sup> tool.

### 3.5.1 Correlation True Positive Experiment

Figure 5 shows the average of 100 separate experiments measuring the true positive rates of an existing, passive timing-based correlation method called IPD-based Correlation<sup>[16]</sup> and watermark-based correlation on FS1 and FS2. The results clearly indicate that IPD-based correlation is vulnerable to even moderate random timing perturbation. 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)).

### 3.5.2 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  $h$ . Figure 6 shows the measured collision rates and expected values are very close, which validates our model.

### 3.5.3 Tradeoff between Watermark Detection Rate and Redundancy Number

Figure 7 shows the average of the measured watermark detection rates of FS1 and FS2 with different redundancy number  $m$ . 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.

## 4 Results and Contributions

The contributions of this research 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 probabilistically robust against random timing perturbations. Our experiments show that our watermark-based correlation is substantially more effective than passive, timing-based correlation in the presence of random timing perturbations. 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. This demonstrates the significant advantage of our active approach over existing passive correlation approaches. 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

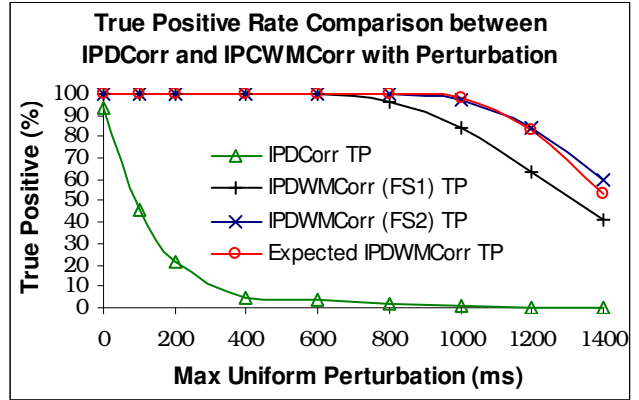


Figure 5. Correlation True Positive Rates under Random Timing Perturbs

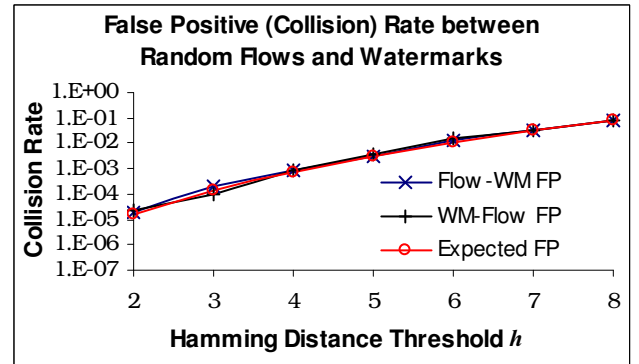


Figure 6. Correlation False Positive (Collision) Rate vs Hamming Distance Threshold  $h$

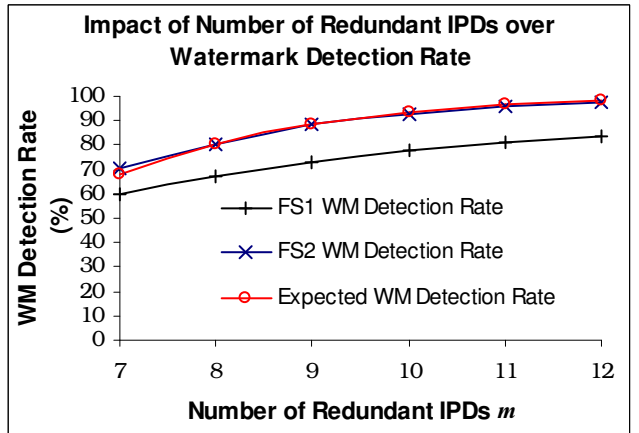


Figure 7. Watermark Detection Rates vs Redundancy Number  $m$

significant practical importance in optimizing the overall correlation effectiveness under real world situations.

## Future Works

This work has become the foundation of the FootFall research project at the Cyber Defense Lab of NC State University, which is generously funded by ARDA ([www.ic-arda.org](http://www.ic-arda.org)). For more information on the continuation of this work, please refer to [footfall.csc.ncsu.edu](http://footfall.csc.ncsu.edu).

## 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 5<sup>th</sup> 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, 7<sup>th</sup> 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, *6<sup>th</sup> 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.