Robust Network-Based Attack Attribution through Probabilistic Watermarking of Packet Flows

Abstract

Network based intruders often 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 encrypted connections through stepping stones, even if those connections are perturbed in timing by the intruder to prevent traceability.

The active watermarking based correlation approach [20] has show advantages over some passive timing based correlation in the presence of active timing perturbation by the attacker. However, the watermarking scheme presented there, based on timing quantization, does not guarantee even adjustment of time over multiple selected packets at real-time. To make the watermark embedding less noticeable to the attacker, it is desirable to adjust the timing evenly over the selected packets.

In this paper, we propose a novel probabilistic watermarking correlation scheme that has guaranteed even timing adjustment over multiple selected packets at real-time. This method has all of the theoretical strength of the provable upper bounds and accurate approximation of the previous quantization based watermarking method [20]. The probabilistic watermarking method essentially trades the watermark embedding success rate for the even time adjustment. Analytical results show that the impact of probabilistic watermarking on watermark detection is equivalent to an additional random delay by the attacker on an otherwise 100% successful watermark embedding scheme. We also identify the provable bounds and accurate tradeoffs between the achievable correlation effectiveness and the defining characteristics of the random timing perturbation. Unlike most previous correlation approaches, our probabilistic watermarking correlation makes no assumptions about the distribution of original inter-packet timing or adversary’s random timing perturbation and it applies to arbitrarily distributed timing perturbation over packet flows with arbitrarily distributed inter-packet timing. Our analytical bounds and tradeoff model hold as long as the random timing perturbation by adversary is bounded. Analytical and experimental results show that the probabilistic watermarking is substantially more robust against random timing perturbations than previous quantization based watermarking, while having virtually the same correlation true positive rate under small timing perturbation.

1 Introduction

Identification of the real source of network based attacks is one of the hardest network security problems due to the various countermeasures the attackers could use to conceal their identities. For example, the attacker could spoof the source IP address of their attack traffic. To trace the attack traffic with spoofed source IP address, various IP traceback techniques [14, 16, 7, 11] have been developed.

Another common and effective way to hide the source of attack traffic is to connect through a sequence of stepping stones [17, 18, 25] 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 this case, the flow of packets from A to B are forwarded by B to C, and become a flow from B to C. The two flows or connections are said to be correlated in this case. The victim at host C can use IP traceback to determine the attack came from host B, but traceback will not be able to determine the attack actually originated from host A. To trace attacks through a stepping stone, it is necessary to correlate incoming traffic with outgoing traffic at the stepping stone.

The earliest work on connection correlation was based on tracking users’ login activities at different hosts [8, 15] . Later work relied on comparing the packet contents, or payloads, of the connections to be correlated [17, 22]. Most recent work has focused on the timing characteristics [21, 24, 25] of connections, in order to correlate encrypted connections (i.e. traffic encrypted using IPSEC [9] or SSH [13, 23]).

While the timing based approach is currently the most capable and promising correlation approach, most existing timing based correlation schemes are vulnerable to active timing perturbation by adversary. For example, the attacker could introduce extra delays at one or more stepping stones to make the correlated flows have very different timing characteristics or to make the uncorrelated flows exhibit similar timing characteristics. To make the timing based correlation robust against active timing perturbation by adversary, Wang and Reeves have recently developed an active timing based correlation scheme [20]. By slightly adjusting the timing of selected packets, they show that 1) an unique watermark could be embedded into encrypted flows; 2) the watermarked flow could be effectively correlated even under active timing perturbation by adversary.

To make the watermark embedding less noticeable to the adversary, it is desirable to evenly distribute the packet timing adjustment (for embedding the watermark) over the selected packets. However, the quantization based watermarking scheme [20] can not guarantee even time adjustment over multiple selected packets at real-time. In this paper, we address the issue of guaranteed even time adjustment over selected packets for active watermarking correlation.

We propose a novel probabilistic watermarking correlation scheme that has guaranteed even time adjustment over multiple selected packets at real time. This method retains the provable properties of the previous quantization based watermarking scheme. In particular, the new probabilistic watermarking scheme can achieve arbitrarily close to 100% correlation true positive rate and arbitrarily close to 0 correlation false positive rate at the same time against arbitrarily large (but bounded) random timing perturbations of arbitrary distribution with arbitrarily small timing adjustment, as long as there are enough packets in the flow.

In addition to even timing adjustment at real-time, the probabilistic watermark embedding method presented in this paper is substantially more robust against timing perturbations than the quantization based scheme. This improved robustness is justified by analysis and demonstrated through experimental results. We also describe the tradeoffs between the parameters of the new method and the achievable correlation performance, and confirm those tradeoffs experimentally.

The remainder of this paper is organized as follows. Section 2 reviews previous work on correlation. Section 3 describes the probabilistic watermarking and analyze the probabilistic watermark bit embedding success rate. Section 4 analyzes the impact of the random timing perturbation over the probabilistic watermarking and identifies the tradeoffs. Section 5 analyzes the overall watermark detection and establishes the theoretical justification of the method. Section 6 experimentally evaluates the probabilistic watermarking scheme and validates the quantitative tradeoff models. Section 7 concludes the paper with possible future research directions.

2 Previous Work

Existing connection correlation approaches are based on three different characteristics: 1) host activity; 2) connection content (i.e., packet payloads); and 3) inter-packet timing. The host activity approach (e.g., CIS [8] and DIDS [15]) 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 [17] and SWT [22]) 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.

Inter-packet timing based correlation approaches (e.g., IPD-based [21], Deviation-based [24] and ON/OFF-based [25]) use the arrival and/or departure times of packets to correlate connections, and they are shown to be effective in correlating encrypted connections.

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. [6] have 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 [6] is the correlation false positive rate.

Wang and Reeves [20] have first identified the accurate quantitative tradeoffs between the achievable correlation effectiveness, the defining characteristics of the random timing perturbation by adversary and the number of packets needed through an active watermarking approach. By embedding a unique watermark with sufficient redundancy into the flow, through adjustment of the timing of selected packets, the active watermarking approach performs significantly better than some passive timing based approach in the presence of active timing perturbation by adversary. It can be made arbitrarily robust to bounded timing perturbations for sufficiently long flows.

Blum et al. [1] have derived provable bounds on the number packets required to passively detect stepping-stone connections with desired false positive rates under the assumption that non-attack streams are sequence of Poisson processes of different rates. However, they have not shown any empirical validation of their basic assumption of the Poisson model of Internet traffic.

While the active watermarking correlation approach has shown its certain advantage over the passive correlation approach in the presence of active timing perturbation by adversary, it is desirable to evenly adjust the timing of selected packets to make the watermarking embedding less noticeable to the adversary. Previous quantization based watermarking scheme [20], however, does not guarantee the even timing adjustment of those selected packets at real-time. In specific, the even timing adjustment of the quantization based watermarking requires the knowledge of arrival times of multiple packets. As the packets of real-time traffic come one by one, the watermark embedding may need to adjust the timing of current packet before the next selected packet arrives. This usually leads to uneven, guessed timing adjustment of selected packets and some packets may need to have long delay to compensate previously guessed, inaccurate delays.

Motivated by the desire to adjust packet timing evenly (and subtly) for watermarking purposes, we propose a novel probabilistic watermarking scheme. This scheme guarantees even timing adjustment for the selected packets, while keeping all the provable strengths of the quantization based watermarking scheme [20].

3 Probabilistic Watermarking

The objective of probabilistic watermarking is to achieve guaranteed even time adjustment on those selected packets at real-time while keeping all the theoretical strength of previous quantization based watermarking scheme [20]. Unlike the quantization based watermarking scheme which has guaranteed 100% watermark embedding success rate, the probabilistic watermarking scheme does not have guaranteed 100% watermark embedding success rate. In other words, the probabilistic watermarking scheme trades off the guaranteed 100% watermark embedding success rate with guaranteed even watermark embedding time adjustment. It is our goal to make the probabilistic watermarking scheme have at least the same watermark detection rate as quantization based watermarking scheme under any timing perturbation level.

We assume the following about random timing perturbations by the attacker:

1. While the attacker can add extra delay to any and all packets of an outgoing flow of any and all stepping stones, the maximum delay he/she could to introduce is bounded.
2. All packets of the incoming flow are kept in their original order, and there are no packets dropped or added by the attacker.
3. While the watermarking scheme is public knowledge, the watermarking embedding and decoding parameters are shared only between the watermark embedder and decoder.

Here we make no assumption about the distribution of the inter-packet timing characteristics of original packet flow, nor we make any assumption about the distribution of the timing perturbation by adversary. The only assumption about the timing perturbation by adversary is that it is bounded.

We now describe the basic method for embedding a watermark in the timing of selected packets of a flow.

3.1 Basic Concept

Given a packet stream $P_1, \ldots, P_n$ with time stamps $t_1, \ldots, t_n$ respectively ($t_i < t_j$ for $1 \leq i < j \leq n$), we first independently and randomly choose $2m$ ($m > 0$) distinct packets: $P_{z_1}, \ldots, P_{z_{2m}}$ ($1 \leq z_k \leq n - d$ for $1 \leq k \leq 2m$), construct $2m$ packet pairs: $\langle P_{z_k}, P_{z_k+d} \rangle$ ($d \geq 1,\ k = 1, \ldots, 2m$).

The IPD (Inter-Packet Delay) between P_{z_k+d} and P_{z_k} is defined as

$$ipd_{z_k,d} \ = \ t_{z_k+d} - t_{z_k},\ (k = 1, \ldots, 2m) \tag{1}$$

Because all P_{z_k} ($k = 1, \ldots, 2m$) are selected independently, $ipd_{z_k,d}$ is independent from each other. Since each P_{z_k} is randomly and independently selected through the same process, $ipd_{z_k,d}$ is identically distributed no matter what inter-packet timing distribution the packet flow $P_1, \ldots, P_n$ may have. Therefore, $ipd_{z_k,d}$ ($k = 1, \ldots, 2m$) is *iid*.

We then randomly divide the $2m$ IPDs into 2 distinct groups of m IPDs. Let $ipd_{1,k,d}$ ($k = 1, \ldots, m$) denote the IPDs in group 1, and let $ipd_{2,k,d}$ ($k = 1, \ldots, m$) denote the IPDs in group 2. Apparently both $ipd_{1,k,d}$ and $ipd_{2,k,d}$ ($k = 1, \ldots, m$) are *iid*. Therefore E($ipd_{1,k,d}$) = E($ipd_{2,k,d}$), and Var($ipd_{1,k,d}$) = Var($ipd_{2,k,d}$).

Let

$$Y_{k,d} \ = \ \frac{ipd_{1,k,d} - ipd_{2,k,d}}{2} \ \ (k = 1, \ldots, m) \tag{2}$$

Then we have E($Y_{k,d}$) = (E($ipd_{1,k,d}$) $-$ E($ipd_{2,k,d}$))/2 = 0. Because $ipd_{1,k,d}$ and $ipd_{2,k,d}$ are *iid*, $Y_{k,d}$ is also (*iid*). We use $\sigma^2_{Y,d}$ to represent the variance.

We represent the average of m $Y_{k,d}$'s as

$$\overline{Y_{m,d}} \ = \ \frac{1}{m} \sum_{k=1}^{m} Y_{k,d} \tag{3}$$

Here we call m the redundancy number. According to the property of variance of independent random variables, we have Var($\overline{Y_{m,d}}$) = $\sigma^2_{Y,d}/m$. Because E($Y_{k,d}$) = 0 ($k = 1, \ldots, m$), E($\overline{Y_{m,d}}$) = 0. Because $Y_{k,d}$ is symmetric ($k = 1, \ldots, m$), $\overline{Y_{m,d}}$ is also symmetric.

To investigate the validity of this concept, we have used tcplib [9] to generate a synthetic telnet flow of 100,000 packet headers, with an empirically derived distribution of telnet packet inter-arrival times. Figure 1 shows the collected histogram of $Y_{k,d}$, for d = 1. It shows that about 89.5% of the IPD differences fall within the range ($-\infty$, 600ms].

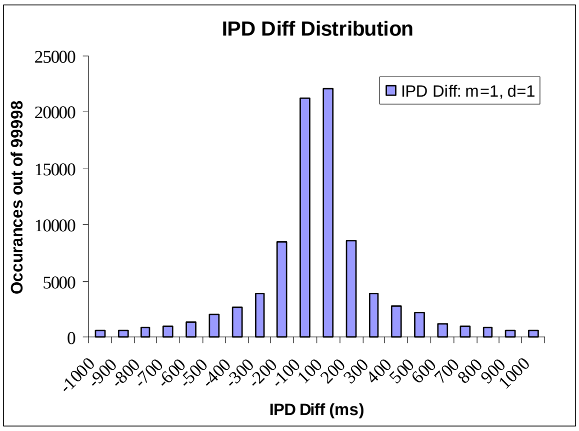

Figure 1: Distribution of $\overline{Y_{m,d}}$ with m = 1 and d = 1

From the 100,000 telnet packets, we generated 99,998 samples of $Y_{k,1}$, as defined in equation (2), from which we calculated $\sigma^2_{Y,1}$ = 8,297,434 (with units of ms). Since the vast majority of adjacent IPDs are less than 10,000ms long, we choose to filter out any IPD that is longer than 10,000 ms in order to reduce the variance $\sigma^2_{Y,1}$. After filtering out any IPD that is longer than 10,000ms [1], we obtained a value of $\sigma^2_{Y,1}$ = 640,063.

[1] We will also filter out those IPDs that are longer than 10,000ms when detecting watermark

3.2 Embedding and Decoding Watermark Bits Probabilistically

We have shown that the distribution of $\overline{Y_{m,d}}$ is symmetric and centered around 0. If we decrease or increase $\overline{Y_{m,d}}$ by an amount $a > 0$, we can skew the distribution to be centered on $-a$ or a, respectively. This increases the probability that $\overline{Y_{m,d}}$ will be negative or positive, and corresponds to a shift of the histogram in Figure 1 to the left or right, respectively. This gives us a way to probabilistically embed a single (binary) watermark bit.

To embed a watermark bit 0, we decrease $\overline{Y_{m,d}}$ by a, so that with probability $>$ 0.5, $\overline{Y_{m,d}}$ will be less than 0. To embed a watermark bit 1, we increase $\overline{Y_{m,d}}$ by a, so that with probability $>$ 0.5, $\overline{Y_{m,d}}$ will be greater than 0. By definition in equation (3), the decrease or increase of $\overline{Y_{m,d}}$ can be easily achieved by decreasing or increasing each of the m $Y_{k,d}$'s by a. By definition in equation (2), the increase of $Y_{k,d}$ by a can be achieved by increasing each $ipd_{1,k,d}$ by a and decreasing each $ipd_{2,k,d}$ by a; the decrease of $Y_{k,d}$ by a can be achieved by decreasing each $ipd_{1,k,d}$ by a and increasing each $ipd_{2,k,d}$ by a.

After $\overline{Y_{m,d}}$ has been decreased or increased by a, we can decode the embedded watermark bit by checking whether $\overline{Y_{m,d}}$ is less than or greater than 0. We interpret the value of $\overline{Y_{m,d}}$ as representing an embedded watermark bit of 1 if it is greater than 0, or representing an embedded watermark bit of 0 if $\overline{Y_{m,d}}$ is less than or equal to 0.

As shown in Figure 1, there is a slight chance such that the embedded watermark (with adjustment $a > 0$) will be decoded incorrectly (ie $\overline{Y_{m,d}} > a$ or $\overline{Y_{m,d}} < -a$). We define the probability that $\overline{Y_{m,d}}$ will be decoded correctly after embedding a watermark bit as the *watermark bit embedding success rate w.r.t. adjustment* a, which can be quantitatively expressed as Pr($\overline{Y_{m,d}} < a$).

The larger the adjustment a is, the higher the watermark bit embedding success rate will be. We now show that even with arbitrarily small $a > 0$, we can achieve arbitrarily close to a 100% watermark bit embedding success rate by having a sufficiently large redundancy number m.

According to the Chebyshev inequality in probability [5], for any random variable X with finite variance Var(X) and for any $t > 0$, $\Pr(\mid X - \mathrm{E(X)} \mid \;\; \geq \; \mathrm{t}) \; \leq \; \mathrm{Var(X)/t^2}$. 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 $\overline{Y_{m,d}}$ with $t = a > 0$, we have

$$\Pr(\mid \overline{Y_{m,d}} \mid \quad \geq \; a) \; \leq \; \frac{\sigma^2_{Y,d}}{ma^2} \tag{4}$$

Because of the symmetry of $\overline{Y_{m,d}}$, we have

$$\Pr(\overline{Y_{m,d}} < a) \; \geq \; 1 - \frac{\sigma^2_{Y,d}}{2ma^2} \tag{5}$$

Equation (5) establishes a lower bound on the probabilistic watermark bit embedding success rate. It indicates that no matter what distribution $Y_{k,d}$ may be, no matter what variance $Y_{k,d}$ may have (as long as it exists), no matter how small the timing adjustment $a > 0$ might be, we can always make the watermark bit embedding success rate arbitrarily close to 100% by increasing the redundancy number m. This result holds true regardless of the distribution of inter-packet timing of the packet flow.

3.3 Analysis on Probabilistic Watermark Bit Embedding Success Rate

We have established a lower bound on the watermark bit embedding success rate through Chebyshev inequality. Now we use the Central Limit Theorem in probability [5] to obtain an accurate approximation to the distribution of the probabilistic watermark bit embedding success rate.

Central Limit Theorem *If the random variables $X_1, \ldots, X_n$ 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*

$$\lim_{n \to \infty} \Pr[\frac{\sqrt{n}(\overline{X_n} - \mu)}{\sigma} \leq x] = \Phi(x) \tag{6}$$

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.

Applying the Central Limit Theorem to random sample $Y_{1,d}, \ldots, Y_{m,d}$, where Var($Y_{k,d}$)= $\sigma^2_{Y,d}$, E($Y_{k,d}$) = 0, we have

$$\Pr[\frac{\sqrt{m}(\overline{Y_{m,d}} - \mathrm{E}(Y_{k,d}))}{\sqrt{\mathrm{Var}(\overline{Y_{m,d}})}} < x] = \Pr[\frac{\sqrt{m}\overline{Y_{m,d}}}{\sigma_{Y,d}} < x] \approx \Phi(x) \tag{7}$$

Therefore

$$\Pr[\overline{Y_{m,d}} < a] = \Pr[\frac{\sqrt{m}\overline{Y_{m,d}}}{\sigma_{Y,d}} < \frac{a\sqrt{m}}{\sigma_{Y,d}}] \approx \Phi(\frac{a\sqrt{m}}{\sigma_{Y,d}}) \tag{8}$$

This means that the distribution of the probabilistic watermark bit embedding success rate is approximately normally distributed with zero mean and variance σ^2/m.

Equation (8) gives us an accurate estimate of the probabilistic watermark bit embedding success rate. For example, with a = 600ms, m = 5 and $\sigma^2_{Y,1}$ = 640,063, Pr[$\overline{Y_{m,1}}$ < 600] $\approx \Phi(\frac{600\sqrt{5}}{\sqrt{640063}}) = \Phi(1.6770) \approx 0.9532$. This means we can expect that the probabilistic watermark bit embedding success rate to be around 0.9532 with a = 600ms, m = 5 and 10 sec filtering. Figure 2 shows the estimation and simulation results of watermark bit embedding success rate for this parameter combination, and various values of m. It demonstrates that equation (8) can give us an accurate estimate of the probabilistic watermark bit embedding success rate for $m \geq 4$ and a = 600ms.

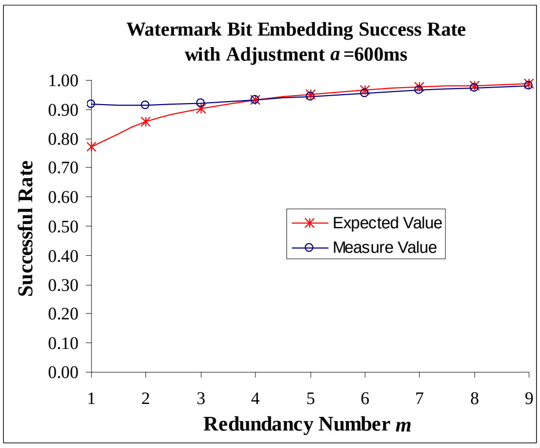

Figure 2: Watermark Bit Embedding Success Rate Estimation and Simulation (a = 600ms, 10 sec. Filtering)

4 Attacker's Impact over Probabilistic Watermark Bit Decoding

We have established the lower bound of and the accurate approximation to the success rate of the probabilistic watermark embedding/decoding. Now we consider the negative impact of the random delay over the probabilistic watermark embedding/decoding.

Let D_i $(i = 1, \ldots, n)$ represent the random delays added to packet P_i by the adversary, let $D > 0$ be the maximum delay the adversary could add to any packet, and let σ_d^2 be the variance of all delays added to all packets. Here we make no assumption about the distribution of random delay the adversary could add to each packet except that the delay is bounded. For example, D_i and D_j $(i \neq j)$ could be correlated to each other and/or have different distributions. This models all the possible bounded random delays the adversary could add to a packet flow.

Given the assumption that the adversary does not know how and which packets are selected by the watermark embedder, the selection of watermark embedding packet P_{z_k} $(k = 1, \ldots, 2m)$ is independent from any random delay D_i the adversary could add. Therefore, the impact of the random delays by adversary over randomly selected P_{z_k} is equivalent to randomly choosing one from the random variable list: $D_1, \ldots, D_n$. Let b_k $(k = 1, \ldots, 2m)$ represent the impact of the random delays by adversary over the k-th randomly selected packet P_{z_k}. Apparently the distribution of b_k is a compound one that depends on the probability that each

D_i would be selected. Since each P_{z_k} is randomly selected according to the same probability distribution over $P_1, \ldots, P_n$, each b_k has the same compound distribution. Furthermore, because each P_{z_k} is selected independently, b_k is also independent from each other. In other words, the impact of any random delays by the adversary over those independently and randomly selected watermark bearing packets is independent and identically distributed (*iid*), and is essentially an *iid* random sample from the random delays the adversary added to all packets.

Let $x_{1,k}$ and $x_{2,k}$ be the random variables that denote the random impact over $ipd_{1,k,d}$ and $ipd_{2,k,d}$ respectively. Apparently both $x_{1,k}$ and $x_{2,k}$ are *iid*. It is also easy to see that $x_{1,k}$, $x_{2,k} \in [-D, D]$, E($x_{1,k}$) = E($x_{2,k}$) = 0, and Var($x_{1,k}$) = Var($x_{2,k}$) = $2\sigma_d^2$. Let X_k = $(x_{1,k} - x_{2,k})/2$, then X_k is *iid*, E(X_k) = 0, and Var(X_k) = σ_d^2.

Let $Y'_{k,d}$ be the random variable that denotes the resulting value of $Y_{k,d}$ after it is perturbed by $x_{1,k}$ and $x_{2,k}$, then we have

$$\begin{aligned} Y'_{k,d} &= [(ipd_{1,k,d} + x_{1,k}) - (ipd_{2,k,d} + x_{2,k})]/2 \\ &= (ipd_{1,k,d} - ipd_{2,k,d})/2 + (x_{1,k} - x_{2,k})/2 \\ &= Y_{k,d} + X_k \end{aligned} \tag{9}$$

Therefore, E($Y'_{k,d}$) = 0. Since $Y_{k,d}$ is *iid* and X_k is *iid*, $Y'_{k,d}$ is also *iid*.

$$\begin{aligned} \text{Var}(Y'_{k,d}) &= \text{Var}(Y_{k,d}) + \text{Var}(X_k) + 2\text{Cov}(Y_{k,d}, X_k) \\ &= \sigma_{Y,d}^2 + \sigma_d^2 + 2\text{Cor}(Y_{k,d}, X_k)\sigma_{Y,d}\sigma_d \\ &\leq \sigma_{Y,d}^2 + \sigma_d^2 + 2\sigma_{Y,d}\sigma_d \\ &= (\sigma_{Y,d} + \sigma_d)^2 \end{aligned} \tag{10}$$

Let $\overline{Y'_{m,d}}$ be the random variable that denotes the resulting value of $\overline{Y_{m,d}}$ after it is perturbed by $x_{1,k}$ and $x_{2,k}$, then we have

$$\overline{Y'_{m,d}} = \frac{1}{m} \sum_{k=1}^{m} Y'_{m,d} \tag{11}$$

According to the property of variance of independent random variables, Var($\overline{Y'_{m,d}}$) = Var($Y'_{k,d}$)/m. It is also easy to see that E($\overline{Y'_{m,d}}$) = 0.

By applying the Chebyshev inequality to $\overline{Y'_{m,d}}$ with $t = a > 0$, we have

$$\text{Pr}(|\ \overline{Y'_{m,d}}\ | \ \geq a) \leq \frac{\text{Var}(Y'_{k,d})}{ma^2} \tag{12}$$

Because of the symmetry of $\overline{Y'_{m,d}}$, we have

$$\text{Pr}(\overline{Y'_{m,d}} < a) \geq 1 - \frac{\text{Var}(Y'_{k,d})}{2ma^2} \geq 1 - \frac{(\sigma_{Y,d} + \sigma_d)^2}{2ma^2} \tag{13}$$

Equation (13) establishes the worst case polynomial lower bound on the probabilistic watermark bit detection rate in the presence of bounded random delays by adversary. It indicates that no matter what inter-packet timing distribution the packet flow may have, no matter what distribution the random timing perturbation by adversary may have, no matter how big the random timing perturbation by adversary may be (as long as it is finite), no matter how small the the watermark embedding adjustment a may be, we can always make the watermark bit detection rate arbitrarily close to 100% by simply increasing the redundancy number m. This result holds true even if the random timing perturbation by adversary is non-*iid*.

By applying the Central Limit Theorem to random sample $Y'_{1,d}, \ldots, Y'_{m,d}$, where Var$(Y'_{k,d}) = \sigma^2_{Y,d} + \sigma^2_d + $ 2Cor$(Y_{k,d}, X_k)\sigma_{Y,d}\sigma_d$, and E$(Y'_{k,d})$ = 0, we have

$$\begin{aligned} \text{Pr}[\frac{\sqrt{m}(\overline{Y'_{m,d}} - \text{E}(Y'_{k,d}))}{\sqrt{\text{Var}(Y'_{k,d})}} < x] &= \text{Pr}[\frac{\sqrt{m}\overline{Y'_{m,d}}}{\sqrt{\text{Var}(Y'_{k,d})}} < x] \\ &= \Phi(x) \end{aligned} \quad (14)$$

Therefore

$$\begin{aligned} \text{Pr}[\overline{Y'_{m,d}} < a] &= \text{Pr}[\frac{\sqrt{m}\overline{Y'_{m,d}}}{\sqrt{\text{Var}(Y'_{k,d})}} < \frac{a\sqrt{m}}{\sqrt{\text{Var}(Y'_{k,d})}}] \\ &\approx \Phi(\frac{a\sqrt{m}}{\sqrt{\text{Var}(Y'_{k,d})}}) \\ &= \Phi(\frac{a\sqrt{m}}{\sqrt{\sigma^2_{Y,d} + \sigma^2_d + 2\text{Cor}(Y_{k,d}, X_k)\sigma_{Y,d}\sigma_d}}) \quad (15) \\ &\geq \Phi(\frac{a\sqrt{m}}{\sigma_{Y,d} + \sigma_d}) \end{aligned}$$

Equation (15) gives us an accurate estimate of the probabilistic watermark bit detection rate in the presence of random delays by the adversary. The correlation coefficient Cor$(Y_{k,d}, X_k)$, whose value range is [-1, 1], models any correlation between the attacker's random delays and the packet timing of the original packet flow. In case the attacker's random delays are independent from the packet timing of the packet flow, Cor$(Y_{k,d}, X_k)$ will be 0.

Comparing equation (15) with equation (8), the impact of the random delays by adversary on probabilistic watermark embedding is as if it increases the variance of $Y_{k,d}$ from $\sigma^2_{Y,d}$ to at most $(\sigma_{Y,d} + \sigma_d)^2$ (equivalently, increases the variance of $\overline{Y_{m,d}}$ from $\sigma^2_{Y,d}/m$ to at most $(\sigma_{Y,d} + \sigma_d)^2/m$). On the other hand, the impact of probabilistic watermarking on the overall watermark detection is as if it increases the variance of the original random delays by the adversary from σ^2_d to at most $(\sigma_{Y,d} + \sigma_d)^2$ on an otherwise 100% successful watermark embedding scheme.

5 Watermark Embedding and Detection

As with symmetric cryptography, the successful detection of embedded watermark requires the knowledge of watermark embedding parameters which is assumed to be a shared secret between the watermark embedder and decoder.

Let the watermark embedding information shared between the watermark embedder and decoder be represented as $\langle S, m, l, a, w \rangle$, where $m \geq 1$ is the redundancy number for embedding one watermark bit, $l > 0$ is the length of the watermark in bits, $a > 0$ is the time adjustment for embedding the watermark, w is the l-bit watermark to be detected and S is the selection function that returns 2 randomly formed groups of m packets each from $2ml$ randomly selected packets. In principle, the more random the packet selection is, the more difficult for the adversary to find out the watermark embedding parameter.

Watermark detection refers to the process of determining if a given watermark is present in a specific connection or flow given the watermark embedding parameters.

Let f denote the flow to be examined and w_f denote the decoded l bits from flow f. The watermark detector works as follows:

1. Decode the l-bit w_f from flow f.

2. Compare the decoded w_f with w.
3. Report that watermark w is detected in flow f if the Hamming distance between w_f and w, represented as $H(w_f, w)$, 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 w is to increase the expected watermark detection rate despite of active countermeasures by adversary. Given any watermark embedding time adjustment $a > 0$, there is always a slight part of distribution of $\overline{Y'_{m,d}}$ that falls outside the range $(-\infty, a]$ (or equivalently $[-a, \infty)$) no matter how many redundant pairs of packets are used. Let $0 < p < 1$ be the probability that $\overline{Y'_{m,d}}$ will fall within range $(-\infty, a]$ (or equivalently $[-a, \infty)$). Then the probability that all l bits are embedded successfully and 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 w_f, the expected watermark detection rate will be

$$\sum_{i=0}^{h} \binom{l}{i} p^{l-i}(1-p)^i \tag{16}$$

For example, for the values p = 0.9532, l = 24, h = 5, the expected watermark detection rate with exact bit match would be p^l = 31.65%. For the same values of p, l, and h, the expected watermark detection rate using a Hamming distance h = 5 would be 99.93%.

It is possible that an unwatermarked flow happens to have packet timing that matches the embedded watermark. In this case, the watermark detector will report the unwatermarked flow as watermarked. It is termed a *collision* between w and f if $H(w_f, w) \leq h$ for an unwatermarked flow f.

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

$$\sum_{i=0}^{h} \binom{l}{i} (\frac{1}{2})^l \tag{17}$$

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 watermarking embedding time adjustment $a > 0$, 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

$$\sum_{i=0}^{h} \binom{l}{i} (\frac{1}{2})^l \leq \sum_{i=0}^{h} \binom{l}{h} (\frac{1}{2})^l \leq (h+1)\frac{l^h}{2^l} \tag{18}$$

Because $\lim_{l\to\infty} \frac{l^h}{2^l} = 0$, we can always make the expected watermark collision probability

$$\sum_{i=0}^{h} \binom{l}{i} (\frac{1}{2})^l < p_c$$

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

$$\sum_{i\,=\,0}^{h} \left(\begin{array}{c} l \\ i \end{array} \right) p^{l-i}(1-p)^{i} \;\geq\; p_d$$

by having $0 < p < 1$ sufficiently close to 1. From inequality (13), this can be accomplished by increasing the redundancy number m regardless of the values of a, $\sigma^2_{Y,d}$ and σ^2_d.

Therefore, in theory, our probabilistic watermarking scheme can 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) random timing perturbation of arbitrary distribution, with arbitrarily small time adjustment of selected packets, as long as there are enough packets in the flow to be watermarked.

Equations (15), (16) and (17) together form the quantitative tradeoff model between the achievable correlation effectiveness (in terms of watermark detection true positive rate and false positive rate), the defining characteristics of the inter-packet timing of the packet flow and the defining characteristics of the random delays by adversary.

6 Experiments

In this section, we empirically validate our probabilistic watermarking scheme. In particular, we seek answers to the following questions:

1. How well does the probabilistic watermarking scheme work compared with the previous quantization based watermarking scheme?
2. What are the implications or impacts of the less than 100% watermark bit embedding success rate of our probabilistic watermarking scheme?
3. How well does our probabilistic watermarking correlation work in the presence of both *iid* and non-*iid* random timing perturbations by adversary?
4. How accurate are our tradeoff models of watermark bit embedding success rate, watermark detection rate and collision rate in predicting both the watermark detection true positive and false positive rates?

We have used two flow sets, labelled FS1 and FS2 in our experiments. FS1 is derived from over 49 million packet headers of the Bell Labs-1 Traces of NLANR [12]. It contains 97 SSH flows that have at least 1000 packets. FS2 contains 1000 telnet flows generated from an empirically-derived distribution [4] of telnet packet inter-arrival times, using the tcplib [3] tool. In our experiments, we have selected one packet out of four packets, and we have chosen to have $d = 1$ and group those selected packets in such a way that $ipd_{1,k,d}$ and $ipd_{2,k,d}$ $(k = 1, \ldots, m)$ are 4 packets away.

We consider two types of timing perturbations in our experiments. The first is *uniformly distributed random perturbation*, in which the attacker at a stepping stone adds to each packet a random delay evenly distributed between 0 and the maximum delay (chosen by the attacker). The second is *batch-releasing perturbation*, in which the attacker at a stepping stone periodically buffers and holds all packets received within a certain time window and forward all the buffered packets in a burst once the time window is over.

Apparently the first type of perturbation is *iid*, and the second type of perturbation is non-*iid*. In fact, the batch-releasing perturbation is neither independent nor identically distributed, whose impact over any packet is closely correlated to the timing of the packet. In addition, batch-releasing drastically changes the original timing characteristics of any flow to a pattern of periodic bursts, which represents a ”tough” case for any timing based correlation.

6.1 Watermark Detection True Positive Experiments

This set of experiments aims to compare the watermark detection rates between the new probabilistic watermarking scheme and existing quantization based watermarking scheme [20] under various perturbations. To ensure a fair comparison, we have chosen the watermark parameters in such a way that the number of IPDs selected and the average time adjustment on those selected IPDs are the same between the two watermarking schemes. For the new probabilistic watermarking scheme, we choose the time adjustment a = 600ms, redundancy number m = 5, Hamming distance threshold h = 5 for 24-bit watermark. For the existing quantization based watermarking scheme, we have chosen the quantization step s = 600ms, redundancy number m = 10, Hamming distance h = 5 for 24-bit watermark. These watermarking parameters would require 240 IPDs selected and would have average 600ms time adjustment for both watermarking schemes.

We first embedded a random 24-bit watermark to each flow in FS1 and FS2. We then randomly perturbed the packet timing of the watermarked flows of FS1 and FS2. It is considered a *true positive* if the embedded watermark can be detected from the timing perturbed watermarked flows. Finally, we calculated the expected watermarking detection rates from equations (15) and (16) for both watermarking schemes under various perturbations.

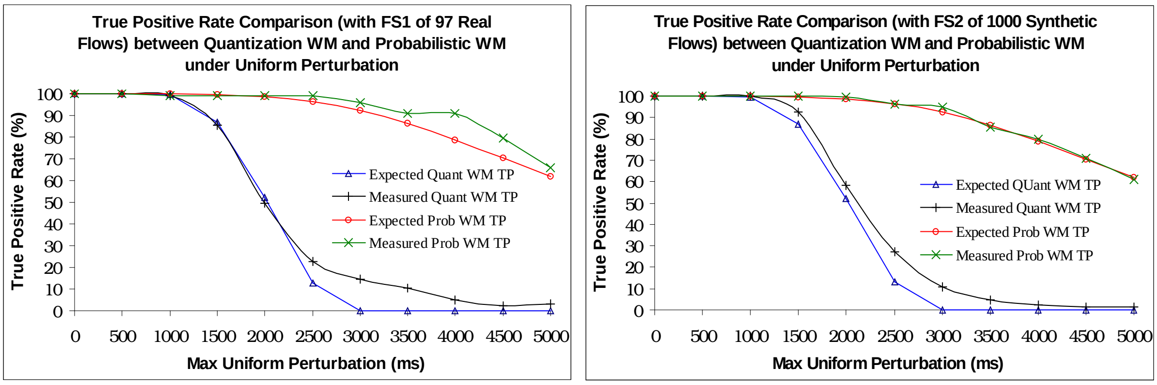

Figure 3: Watermark detection true positive rates under a uniform random timing perturbation, for a) 97 real flows (FS1) and b) 1000 synthetic flows (FS2)

Figure 3 shows the expected and measured watermark detection rates for both probabilistic watermarking and quantization based watermarking schemes on FS1 and FS2 under various uniformly distributed random timing perturbations. For any uniformly distributed timing perturbation less than 1000ms, both the quantization based watermarking and the probabilistic watermarking schemes have very close to 100% watermark detection rate. As the timing perturbation level increases, both the expected and measured watermark detection rates for quantization based watermarking scheme drop quickly. On the other hand, the expected and measured watermark detection rates for the probabilistic watermarking scheme appear much more robust against large timing perturbations. For example, at 2000ms maximum uniform perturbation, the true positive rate for the quantization based watermarking scheme drops to around 50%, and the true positive rate for the probabilistic watermarking scheme remains at 99%. Because the uniformly distributed random timing perturbation is independent from the original timing characteristics of the packet flow, the expected watermark detection rates are calculated in such a way that Cor($Y_{k,d}, X_k$) = 0. Figure 3 shows that the measured probabilistic watermark detection rates under various levels of uniformly distributed perturbation are well approximated by estimated values derived from equation (15).

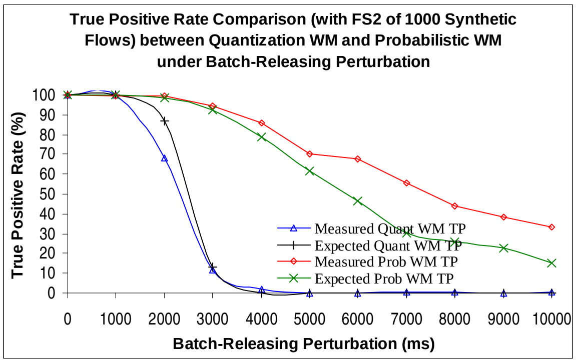

Figure 4: Watermark detection true positive rates under batch release timing perturbations, for 1000 synthetic flows (FS2)

Figure 4 shows both the expected and measured watermark detection rates for both probabilistic watermarking and quantization based watermarking schemes on FS2 under various levels of batch-releasing timing perturbations. The expected true positive rates are calculated with Cor($Y_{k,d}, X_k$) = 0. For large perturbations, the measured probabilistic watermarking detection rates are better than expected, and for some small perturbations, the measured quantization based watermarking detection rates are a little less than expected. Again, the probabilistic watermarking scheme is substantially more robust against large batch-releasing timing perturbations than the quantization based watermarking scheme. This experiment demonstrates that the probabilistic watermarking scheme works even if the timing perturbation is non-*iid*.

In theory, the probabilistic watermarking scheme has less than 100% watermark bit embedding success rate, and it has less than 100% expected watermark detection rate even without any perturbation. This means that the probabilistic watermarking scheme would have slightly lower watermark detection rate than the quantization based watermarking scheme without any timing perturbation. Our experimental results show that while the probabilistic watermarking scheme appears more sensitive to how the selected packets are grouped, it could have watermark detection rates comparable to that of quantization based watermarking scheme under small timing perturbations. Under large timing perturbation, the probabilistic watermarking scheme has substantially higher watermark detection rate than the quantization based watermarking scheme.

6.2 Watermark Detection False Positive Experiments

No matter how the watermark is chosen, it is possible that the packet timing of an unwatermarked flow will cause our analysis method to report the detection of the watermark. This occurrence is termed a watermark *collision*, or watermark detection *false positive.* According to our watermark detection false positive model (17), the collision probability is determined by the number of watermark bits l and the Hamming distance threshold h.

We experimentally investigated the collision rate between a given flow and 10,000∼1,000,000 randomly generated 24- bit watermarks. Figure 5 shows the expected and measured watermark detection false positive rates under various Hamming distance thresholds. The measured values are very close to the expected values, which validates our assumption that the l-bit w_f extracted from random flow f is uniformly distributed. The experimental results also confirm our analytical result that arbitrary close to 0 watermark collision probability can be achieved by a sufficiently large number of watermark bits l and smaller Hamming distance threshold h.

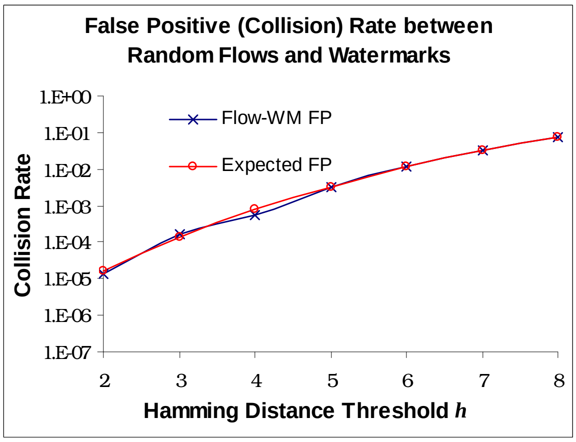

Figure 5: Watermark detection false positive rate vs. the threshold h

6.3 Watermark Detection Tradeoff Experiments

Equation (15) gives us the quantitative tradeoff between the watermark embedding time adjustment a, the redundancy number m, the defining characteristics of the random timing perturbation and the expected watermark embedding detection rate p. Given any watermark embedding detection rate p, equation (16) gives us the tradeoff between the number of watermark bits m, the Hamming distance threshold h and the expected watermark detection rate.

To verify the validity and accuracy of our model of the watermark detection rate, we first embedded a random 24-bit watermark into each flows in FS1 and FS2 with different watermark embedding time adjustment a, redundancy number m, we then perturbed those watermarked flows with up to 5000ms uniformly distributed

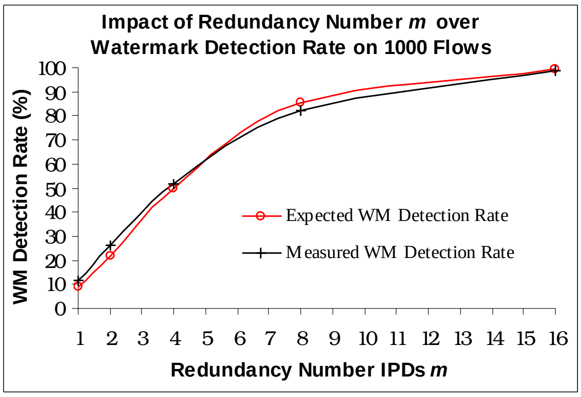

Figure 6: Watermark detection true positive rates under 5000ms uniformly distributed perturbation vs. the redundancy number m

random timing perturbation. We finally attempted to detect the embedded watermark from those perturbed watermarked flows.

Figure 6 shows both expected and measured values of the watermark detection rate and the redundancy number used in embedding the watermark. The 24-bit watermark was embedded with embedding time adjustment a = 600ms, different redundancy number m = 1,2,4,8,16, and was detected with Hamming distance threshold h = 5. The measured values are very close to the expected values based on our tradeoff models. In particular, under 5000ms uniformly distributed random perturbation, embedding watermark with m = 1 only achieves about 10% watermark detection rate. With m = 4, the watermark detection rate increases to about 50%. Once m is increased to 16, the watermark detection rate is increased to 98.9% under the same level of 5000ms uniformly distributed random timing perturbation. These results empirically confirm our analytical conclusion that we can always achieve arbitrarily close to 100% watermark detection rate by increasing the redundancy number m no matter how big the random timing perturbation may be.

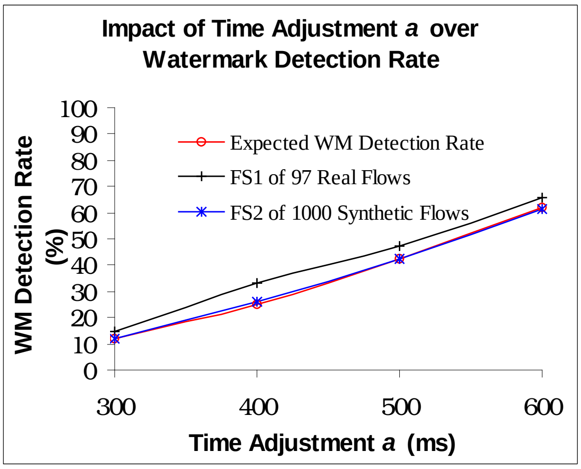

Figure 7: Watermark detection true positive rates under 5000ms uniformly distributed perturbation vs. the timing adjustment a

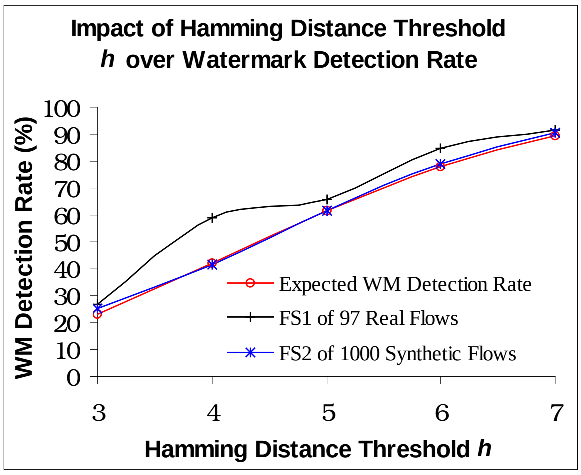

Figure 8: Watermark detection true positive rates under 5000ms uniformly distributed perturbation vs. the hamming distance threshold h

Figure 7 shows both the expected and measured tradeoffs between the watermark detection rate and the watermark embedding time adjustment a used in embedding the watermark. The 24-bit watermark was embedded with embedding time adjustment a = 300, 400, 500, 600ms, redundancy number m = 5, and was detected with Hamming distance threshold h = 5. The measured values for FS1 are close to the expected values, and the measured values for FS2 are extremely close to the expected values.

Figure 8 shows both the expected and measured tradeoffs between the watermark detection rate and the Hamming distance threshold h. The 24-bit watermark was embedded with embedding time adjustment a = 600ms, redundancy number m = 5, and was detected with different Hamming distance threshold h = 3,4,5,6,7. Again, the measured values for FS1 is generally close to the expected values, and the measured values for FS2 are almost identical to the expected values.

In summary, these experimental results not only validate our tradeoff models but also confirm our hypothesis that random variable $Y_{k,d}$ defined over the randomly selected IPDs out of non-*iid* IPDs could be *iid*. Therefore, our analytical results hold even if the original IPDs are not *iid*.

7 Conclusions and Future Work

Motivated by solving the uneven real-time adjustment problem in the quantization based watermarking scheme [20], we presented a probabilistic watermarking scheme that 2) has perfect even real-time adjustment; 2) keeps all the theoretical strength of provable upper bounds and accurate approximation of the quantization based watermarking scheme.

We essentially traded the watermark bit embedding success rate for the guaranteed even time adjustment. As a result, the probabilistic watermarking scheme has less than 100% watermark embedding success rate. Both our analytical and experimental results show that the probabilistic watermarking scheme is substantially more robust against large time perturbations than the quantization based watermarking. As for the small time perturbation, the probabilistic watermarking scheme has slightly lower but comparable watermark detection rate than the quantization based watermarking scheme. By grouping the random selected IPDs carefully, the probabilistic watermarking scheme can have virtually the same watermark detection rate under small timing perturbation and have much higher watermark detection rate under large timing perturbation.

We developed models of tradeoffs between the achievable watermarking detection rate, collision rate, watermarking embedding parameters and the defining characteristics of the random timing perturbation. In addition, we identified 1) the provable upper bound on the number of the packets to achieve certain level of correlation effectiveness under certain level of random timing perturbation; 2) the provable lower bound on the watermark correlation effectiveness given the number of packets and the level of random timing perturbation. Unlike most previous correlation approaches, our bounds and models make no assumptions about the distribution of original inter-packet timing of the packet flow or distribution of the random timing perturbation by adversary. This makes our analytical framework applies to packet flows of arbitrary inter-packet timing characteristics perturbed with arbitrarily distributed random timing perturbation. Our experimental results validate that our probabilistic watermarking scheme works in the presence of both *iid* and non-*iid* random timing perturbation.

It is an area of future work to address packet flow correlation problem when the adversary introduces both chaff (bogus packets) and random timing perturbation are at the same time.

References

[1] A. Blum, D. Song, and S. Venkataraman. Detection of Interactive Stepping Stones: Algorithms and Confidence Bounds. In *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004)*. Springer, October 2004.

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

[3] P. Danzig and S. Jamin. Tcplib: A Library of TCP Internetwork Traffic Characteristics. Technical Report USC-CS-91-495, University of Southern California, 1991.

[4] P. Danzig, S. Jamin, R. Cacerest, D. Mitzel, and E. Estrin. An Empirical Eorkload Model for Driving Wide-Aea TCP/IP Network Simulations. *Journal of Internetworking*, 3(1):1–26, March 1992.

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

[6] D. Donoho. et al. 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): LNCS-2516*, pages 17–35. Springer, October 2002.

[7] M. T. Goodrich. Efficient packet marking for large-scale ip traceback. In *Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS 2002)*, pages 117–126. ACM, October 2002.

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

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

[10] G. Kramer. Generator of Self-Similar Network Traffic. URL. http://wwwcsif.cs.ucdavis.edu/ kramer/code/trf_gen2.html.

[11] J. Li, M. Sung, J. Xu and L. Li. Large Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, IEEE, 2004.

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

[13] OpenSSH. URL. http://www.openssh.com.

[14] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. In *Proceedings of ACM SIGCOMM 2000*, pages 295–306. ACM, September 2000.

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

[16] A. Snoeren, C. Patridge, et. al. Hash-based IP Traceback. In *Proceedings of ACM SIGCOMM 2001*, pages 3–14. ACM, September 2001.

[17] S. Staniford-Chen and L. Heberlein. Holding Intruders Accountable on the Internet. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pages 39–49. IEEE, 1995.

[18] C. Stoll. *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage.* Pocket Books, 2000.

[19] M. S. Taqqu, W. Willinger, and R. Sherman. Proof of a Fundamental Result in Self-Similar Traffic Modeling. *ACM Computer Communication Review*, 27:5–23, 1997.

[20] X. Wang and D. Reeves. Robust Correlation of Encrypted Attack Traffic through Stepping Stones by Manipulation of Interpacket Delays. In *Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS 2003)*, pages 20–29. ACM, October 2003.

[21] X. Wang, D. Reeves, and S. Wu. Inter-packet Delay based Correlation for Tracing Encrypted Connections through Stepping Stones. In *Proceedings of the 7th European Symposium on Research in Computer Security (ESORICS 2002), LNCS-2502*, pages 244–263. Springer-Verlag, October 2002.

[22] X. Wang, D. Reeves, S. Wu, and J. Yuill. Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework. In *Proceedings of the 16th Internatinal Conference on Information Security (IFIP/Sec 2001)*, pages 369–384. Kluwer Academic Publishers, June 2001.

[23] T. Ylonen and C. Lonvick. *IETF Internet Draft: SSH Protocol Architecture*. IETF, June 2004. draft-ietf-secsh-architecture-16.txt, work in progress.

[24] K. Yoda and H. Etoh. Finding a Connection Chain for Tracing Intruders. In *Proceedings of the 6th European Symposium on Research in Computer Security (ESORICS 2000), LNCS-1895*, pages 191–205. Springer-Verlag, October 2002.

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