Robust Correlation of Encrypted Attack Traffic through Stepping Stones by Flow Watermarking

Xinyuan Wang, *Member, IEEE,* Douglas S. Reeves, *Member, IEEE*

***Abstract*— Network based intruders seldom attack their victims directly from their own computer. Often, they stage their attacks through intermediate "stepping stones" in order to conceal their identity and origin. To identify the source of the attack behind the stepping stone(s), it is necessary to correlate the incoming and outgoing flows or connections of a stepping stone. To resist attempts at correlation, the attacker may encrypt or otherwise manipulate the connection traffic.**

Timing based correlation approaches have been shown to be quite effective in correlating encrypted connections. However, timing based correlation approaches are subject to timing perturbations that may be deliberately 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. Unlike most previous timing based correlation approaches, our watermark-based approach is "active" in that it embeds a unique watermark into the encrypted flows by slightly adjusting the timing of selected packets. The unique watermark that is embedded in the encrypted flow gives us a number of advantages over passive timing based correlation in resisting timing perturbations by the attacker. In contrast to existing passive correlation approaches, our active watermark based correlation does not make any limiting assumptions about the distribution or random process of the original inter-packet timing of the packet flow. In theory, our watermark based correlation can achieve arbitrarily close to 100% correlation true positive rate and arbitrarily close to 0% false positive rate at the same time for sufficiently long flows, despite arbitrarily large (but bounded) timing perturbations of any distribution by the attacker. Our work is the first that identifies 1) accurate quantitative tradeoffs between the achievable correlation effectiveness and the defining characteristics of the timing perturbation; 2) a provable upper bound on the number of packets needed to achieve a desired correlation effectiveness, given the amount of timing perturbation.

Experimental results show that our active watermark based correlation performs better and requires fewer packets than existing, passive timing based correlation methods in the presence of random timing perturbations.

***Index Terms*— Network-level security and protection, intrusion tracing, correlation, stepping stone.**

I. Introduction

NETWORK based attacks have become a serious threat to the critical information infrastructure on which we depend. To stop or repel network-based attacks, it is critical to be able to identify the source of the attack. Attackers, however, go to some lengths to conceal their identities and origin, using a variety of countermeasures. As an example, they may spoof the IP source address of the attack traffic. Methods of tracing spoofed traffic, generally known as IP traceback [23], [26], [9], [14] have been developed to address this countermeasure.

Another common and effective countermeasure used by network-based intruders to hide their identity is to connect through a sequence of intermediate hosts, or *stepping stones*, 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, the incoming packets of an attack connection from A to B are forwarded by B, and become outgoing packets of a connection from B to C. The two connections or flows are related in such a case. The victim host C can use IP traceback to determine the second flow originated from host B, but traceback will not be able to correlate that with the attack flow originating from host A. To trace attacks through a stepping stone, it is necessary to correlate the incoming traffic with the 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 user's login activities at different hosts [11], [25]. Later work relied on comparing the packet contents, or payloads, of the connections to be correlated [27], [33]. Most recent work has focused on the timing characteristics [36], [35], [32], [8], [1] of connections, in order to correlate encrypted connections (i.e. traffic encrypted using IPSEC [12] or SSH [18], [34]).

Timing based correlation approaches, however, are sensitive to the use of countermeasures by the attacker, or adversary. 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 stones. This kind of timing perturbation will adversely affect the effectiveness of any timing-based correlation. Timing perturbation can either make unrelated flows have similar timing characteristics, or make related flows exhibit different timing characteristics. This will increase the correlation false positive rate, or decrease the correlation true positive rate, respectively.

Previous timing-based correlation approaches are passive in that they simply examine (but do not manipulate) the traffic timing characteristics for correlation purposes. While passive approaches are simple and easy to implement, they may be vulnerable to active countermeasures by the attacker, and/or require a large number of packets in order to correlate timing-perturbed flows.

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

We propose a novel watermark-based correlation scheme that is designed specifically to be robust against timing perturbations by the adversary. Unlike most previous correlation approaches, our watermark-based approach is *active*; that is, it embeds a unique

Xinyuan Wang is with the Department of Computer Science, George Mason University, Fairfax, VA 22030, USA. email: xwangc@gmu.edu

Douglas S. Reeves is with the Departments of Computer Science, Electrical and Computer Engineering, N. C. State University, Raleigh, NC 27695, USA. email: reeves@eos.ncsu.edu

watermark into the encrypted flows by slightly adjusting the timing of selected packets. The unique watermark that is embedded in the encrypted flow gives us a number of advantages over passive timing based correlation in overcoming timing perturbations by the adversary. First, our active watermark based correlation does not make any limiting assumptions about the distribution or random process of the original inter-packet timing of the packet flow, or the distribution of random delays an adversary can add. This is in contrast to existing passive timing based correlation approaches. Second, our method requires substantially fewer packets in the flow to achieve the same level of correlation effectiveness as existing passive timing based correlation. In theory, our watermark based correlation can achieve arbitrarily close to 100% correlation true positive rate and arbitrarily close to 0% false positive rate at the same time for sufficiently long flows, despite arbitrarily large (but bounded) timing perturbation of arbitrary distribution by the adversary. To the best of our knowledge, our work is the first that identifies 1) the accurate quantitative tradeoffs between the achievable correlation effectiveness and the defining characteristics of the timing perturbation; 2) a provable upper bound on the number of packets needed to achieve a desired correlation effectiveness, given a bound on the amount of timing perturbation.

We also investigate the maximum negative impact on the embedded watermark an adversary can have, and the minimum effort needed to achieve that impact. Under the condition that the watermark embedding parameters are unknown to the adversary, we determine the minimum distortion required for the adversary to completely eliminate any embedded watermark from the inter-packet timing, and the optimal strategy for doing so. We further investigate the implications of the constraints of real-time communication and bounded delay for the adversary's ability to remove the embedded watermark. While there exist ways to completely eliminate hidden information from any signal offline, we show that (without knowledge of the watermark embedding parameters) it is generally infeasible for the adversary to completely eliminate the embedded watermark from the packet timing in real-time, even if he can introduce arbitrarily large (but bounded) distortion to the packet timing of normal network traffic. This result ensures that our watermark-based correlation is able to withstand arbitrarily large timing perturbations in real-time, provided there are enough packets in the flows to be correlated.

The remainder of this paper is organized as follows. Section II summarizes related work. Section III gives an overview of watermark-based correlation. Section IV describes the embedding of a single watermark bit in a flow. Section V analyzes the watermark bit robustness, tradeoffs, and the overall watermark detection and collision rates. Section VI analyzes the minimum distortion required to completely remove an arbitrary embedded watermark, the optimal strategy for doing so, and the implications of real-time constraints. Section VII describes the implementation of our method in the Linux kernel, and evaluates the effectiveness of out method empirically. Section VIII concludes the paper, and points out some future research directions.

II. Related Work

Existing connection correlation approaches are based on three different characteristics: 1) host activity; 2) connection content (i.e. packet payload); and 3) inter-packet timing characteristics. The host activity based approach (e.g. DIDS [25] and CIS [11]) collects and tracks users' login activity at each stepping stone. The major drawback of host activity based methods is that the host activity collected from each stepping stone is generally not trustworthy. Since the attacker is assumed to have full control over each stepping stone, he/she can easily modify, delete or forge user login information. This defeats the ability to correlate based on host activity.

Content based correlation approaches (e.g. Thumbprinting [27] and SWT [33]) require that the payload of packets remains invariant across (i.e., is unchanged by) stepping stones. Since the attacker can easily transform the connection content by encryption at the application layer, these approaches are suitable only for unencrypted connections.

To correlate encrypted traffic, timing based approaches (e.g. ON/OFF-based [36], Deviation-based [35] and IPD-based [32]) have been proposed. These methods passively monitor the arrival and/or departure times of packets, and use this information to correlate incoming and outgoing flows of a stepping stone. For instance, IPD-based correlation [32] has shown that 1) the important inter-packet timing characteristics of connections are preserved during transit across many routers and stepping stones; and 2) the timing characteristics of interactive flows (e.g.telnet and SSH connections) are almost always unique enough to differentiate related flows from unrelated flows.

While the first generation of timing based correlation approaches have proved to be effective in correlating encrypted connections, they are vulnerable to the attacker's use of active timing perturbation. Donoho et al. [8] first investigated the theoretical limits on the attacker's ability to disguise his traffic through timing perturbation and bogus (padding, or *chaff*) packet injection. By using multiple-timescale analysis techniques, they show that correlation based on long term behavior (of sufficiently long flows) is still possible despite certain timing perturbations by the attackers. 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 not addressed by [8] is the correlation false positive rate. While coarse timescale analysis for long term behavior may not be affected by packet jitter (timing perturbations) introduced by the attackers, it may also be insensitive to the unique details of each flow's timing. Therefore coarse scale analysis tends to increase the correlation false positive rate, while increasing the correlation true positive rate of timing-perturbed flows. Nevertheless, Donoho et al's work [8] represents an important first step towards understanding the inherent limitations of timing perturbations by the adversary on timing-based correlation. Not addressed in this work were questions about correlation effectiveness for flows of arbitrary (rather than Poission or Pareto) distribution in packet timing, and the achievable tradeoff of false and true positive rates given the magnitude of the timing perturbation and the number of packets available.

After the initial publication of our active watermarking based correlation [31], Blum et al. [1] developed another passive, timing based correlation method that considers both correlation true positive and false positive at the same time. Based on the assumption that the inter-packet timing of flows can be modelled as a sequence of Poisson processes of different rates, they derived upper bounds on the number of packets needed to achieve a specified false positive rate and a 100% true positive rate. They also derived the lower bounds on the amount of chaff needed to

defeat their passive method of correlation. However, their work did not present any experimental results, nor did it address such practical issues as how to derive model parameters in real-time, or how many packets are needed in practice for real flows and realistic timing perturbations. Zhang et al. [37] and He et al. [10] recently proposed several new passive timing based correlation methods based on similar assumptions used by Blum et al. [1], and they showed that their methods have better performance than what proposed by Blum et al. [1].

Chakinala et al. [2] formally analyzed the packet reordering channels as information-theoretic game. Peng et al. [19] recently studied the secrecy of the watermark based correlation [31] and proposed an offline statistical method for detecting the existence of watermark from a packet flow. However, their method assumes the watermark embedding follows some simple and fixed patterns and requires access to both the unwatermarked and watermarked flows to be effective. As we will show in the watermark tracking model, we can make the unwatermarked flow unavailable to the adversary by watermarking the packet flow from its source.

In summary, existing timing-based correlation approaches passively measure and use possibly perturbed timing information for correlation. They do not attempt to make the inter-packet timing characteristics more amenable to effective correlation. Passive approaches are simple to implement and undetectable by the attacker. However, they generally make more limiting assumptions about the inter-packet timing characteristics, and require more packets than an active approach to effectively correlate timing-perturbed flows, as we will show.

In this paper, we describe a novel active timing-based correlation approach that 1) makes no assumption about the distribution of inter-packet timing intervals; 2) does not require the timing perturbation to follow any specific distribution or random process; 3) is provably effective against certain correlated random timing perturbation; and 4) requires substantially fewer packets than passive approaches to achieve the same level of correlation effectiveness.

III. Overview of Watermark-Based Correlation

A. Overall Watermark Tracing Model

The watermark tracing approach exploits the observation that interactive connections (i.e. Telnet, SSH) are bidirectional. The idea is to watermark the backward traffic (from victim back to the attacker) of the bidirectional attack connections by slightly adjusting the timing of selected packets. If the embedded watermark is both robust and unique, the watermarked back traffic can be effectively correlated and traced across stepping stones, from the victim all the way back to the attacker. As shown in Figure 1, the attacker may connect through a number of hosts $(H_1, \ldots, H_n)$ before attacking the final target. Assuming the attacker has not gained full control on the attack target, the attack target will initiate the attack tracing after it has detected the attack. Specifically, the attack target will watermark the backward traffic of the attack connection, and inform sensors across the network about the watermark. The sensors across the network will scan all traffic for the presence of the indicated watermark, and report to the target if any occurrences of the watermark are detected.

Gateway, firewall and edge router are good places to deploy sensors. However, how many sensors can be deployed depend on not only the resources available but also the administrative privilege. How to optimally deploy limited number of sensors over particular network is an open research problem [22]. Due to space limitation, we leave aside the sensor deployment issues, and instead focus on the watermark tracing approach itself.

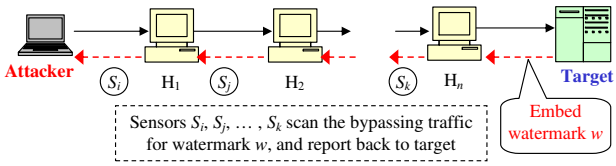

Fig. 1. Overall Watermark Tracing Model

Since the backward traffic is watermarked at its very source – the attack target, which is not controlled by the attacker, the attacker will not have access to an unwatermarked version of the traffic. This makes it difficult for the attacker to determine which packets have been delayed by the watermarking process, running at the target.

The objective of watermark-based correlation is to make the correlation of encrypted connections probabilistically robust against random timing perturbations by the adversary. Unlike existing timing-based correlation schemes, our watermark-based correlation is active in that it embeds a unique watermark into the encrypted flows, by slightly adjusting the timing of selected packets. If the embedded watermark is both unique and robust, the watermarked flows can be effectively identified and thus correlated at each stepping stone.

In contrast to most previous passive correlation approaches, our watermark-based correlation makes no limiting assumption about the distribution or random process of the original inter-packet timing characteristics of the flows to be correlated.

We assume the following about the random timing perturbations introduced by the adversary:

1) While the attacker can add extra delay to any or all packets of an outgoing flow at the stepping stone, the maximum delay he or she can introduce is bounded.
2) All packets in the original flow are kept. No packets are dropped from or added to the flow by the stepping stone.
3) While the watermarking scheme is public knowledge, the watermarking embedding and decoding parameters are secrets known only to the watermark embedder and the watermark detector(s).

Here we do not require that the packet order of two flows be the same, as long as the total number of packets is not modified. As shown in works [20], [21], [30], our watermark-based approach is able to correlate encrypted flows even if chaff and timing perturbation are applied at the same time. Due to space limitation, we only consider timing perturbation in this paper.

In contrast to all previous passive approaches, our correlation method does not require the random timing perturbation introduced by the attacker to follow any particular distribution or random process to be effective. The only assumption about timing perturbations is that they follow some distribution of finite variance and they have the same covariance among each other.

B. Watermarking Model and Concept

Generally, digital watermarking [4] involves the selection of a watermark carrier, and the design of two complementary processes: embedding and decoding. The watermark embedding process inserts the watermark information into the carrier signal

by a slight modification of some property of the carrier. The watermark decoding process detects and extracts the watermark (equivalently, determines the existence of a given watermark) from the carrier signal. To correlate encrypted connections, we propose to use the inter-packet timing as the watermark carrier property of interest.

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 ith 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 queueing 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

$$ipd_{i,j} = t_j - t_i \tag{1}$$

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

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

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 $ipd_{i,j}$: $ipd'_{i,j} - ipd_{i,j} = d_j - d_i$. Note we use the timestamp of the ith and the jth packets to calculate $ipd_{i,j}$ or $ipd'_{i,j}$ even if there might be some packets reordered in the packet flow. Since we only use the timestamp of selected packets, the negative impact of using the "wrong" packet due to packet reorder is equivalent to some random timing perturbation over the IPD.

Assume $D > 0$ is the maximum delay that the attacker can add to P_i $(i = 1, \ldots, n)$, 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 perturbations by the adversary, we choose to embed the watermark using IPDs from randomly and independently selected packets.

Given a flow containing the packet sequence $P_1, \ldots, P_n$ with time stamps $t_1, \ldots, t_n$ respectively ($t_i < t_j$ for $1 \le i < j \le n$), we can independently and probabilistically choose $2m < n$ packets through the following process: (1) sequentially consider each of the n packets; and (2) independently and randomly determine if the current packet will be chosen for watermarking purposes, with probability $p = \frac{2m}{n}$ $(0 < m < \frac{n}{2})$. By this method, the selection of one packet for watermarking purposes is independent from the selection of any other packet. Therefore, we can expect to have $2m$ distinct packets independently and randomly selected from a packet stream of n packets.

IV. Basic and Probabilistic Watermarking

A. 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

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

where round(x) is the function that rounds off real number x to its nearest integer (i.e. round(x)=i for any $x \in ([i-0.5,\ i+0.5))$.

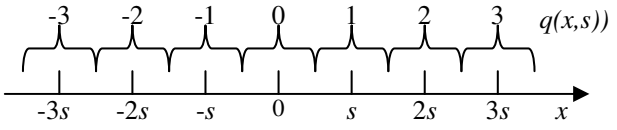

Fig. 2. Quantization of Scalar Value x

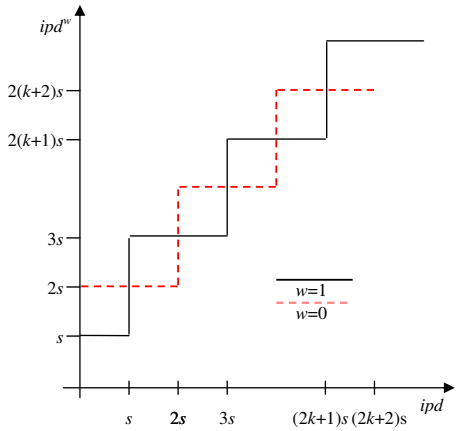

Fig. 3. Mapping between Unwatermarked *ipd* and Watermarked ipd^w to Embed Watermark Bit w

Figure 2 illustrates the quantization for 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 digit or 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 digit w, the watermark bit embedding is defined as function

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

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

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*, i.e., packets can be delayed, but cannot be output earlier than they arrive. Figure 3 illustrates the embedding of watermark bit w by mapping ranges of unwatermarked *ipd* to the corresponding watermark ipd^w.

The watermark bit decoding function is defined as

$$d(ipd^w, s) = q(ipd^w, s) \mod 2 \tag{5}$$

The correctness of watermark embedding and decoding is guaranteed by the following theorems, whose proofs are omitted due to space limitation.

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

B. 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 $\Delta_{\max}$ to ipd will not change the result of watermark decoding ($d(ipd, s)$), while a perturbation of $\Delta_{\max}$ or greater to ipd may change the watermark decoding result.

We define the *tolerable perturbation range* as the portion 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 range of perturbation values outside the tolerable perturbation range.

Given any $ipd > 0, s > 0$ and binary watermark bit w, by definition of quantization function $q(ipd, s)$ in (3) and watermark decoding function $d(ipd^w, s)$ 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 corrupted by the timing perturbation. 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 the watermarked flow more, as the watermark bit embedding itself may add up to $2s$ 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 remainder of this 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 can make the embedded watermark bit robust, with arbitrarily high probability, against arbitrarily large (and yet bounded) random timing perturbation by the attacker, as long as the flow to be watermarked contains a sufficient number of packets.

C. 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 assumption that the attacker does not know which packets are randomly selected and which IPDs will be used for embedding the watermark.

We apply two strategies to contain and minimize the impact of random delays over the watermark-bearing IPDs. The first strategy is to distribute watermark-bearing IPDs over a longer duration of the flow. The second is to embed 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 one watermark bit in one IPD, we embed the watermark bit into the average of $m \geq 1$ randomly selected IPDs. Here we call m the *redundancy number*.

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 $2m$ ($0 < m < \frac{n}{2}$) distinct packets: $P_{x_1}, \ldots, P_{x_{2m}}$ ($1 \leq x_k \leq n$ for $1 \leq k \leq 2m$). We then randomly divide the $2m$ packets into two groups of m packets $\{P_{y_1}, \ldots, P_{y_m}\}$ and $\{P_{z_1}, \ldots, P_{z_m}\}$ ($y_k < z_k$ and $y_k, z_k \in \{x_1, \ldots, x_{2m}\}$), and randomly form m packet pairs $\{< P_{y_1}, P_{z_1} >, \ldots, < P_{y_m}, P_{z_m} >\}$.

Let $< P_{y_k}, P_{z_k} >$ be the k-th pair of packets randomly selected to embed the watermark bit, whose timestamps are t_{y_k} and t_{z_k} respectively. Then we have m IPDs: $ipd_k = t_{z_k} - t_{y_k} (k = 1, \ldots, m)$. We represent the average of these m IPDs as

$$ipd_{avg} = \frac{1}{m} \sum_{k=1}^{m} ipd_k \tag{6}$$

Given any 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 equation (4) to ipd_{avg}. Specifically, the timing of packets P_{z_k} ($k = 1, \ldots, m$) are all delayed so that ipd_{avg} is adjusted by Δ, as defined in equation (4). To decode the watermark bit, we first collect the m IPDs (denoted as $ipd_k^w, k = 1, \ldots, m$) from the same m pairs of randomly selected packets and from them compute the average ipd_{avg}^w of $ipd_1^w, \ldots, ipd_m^w$. Then we can apply the decoding function defined in equation (5) to ipd_{avg}^w to decode the watermark bit.

Because watermark embedding is now applied to the average of m IPDs, the watermark embedding process needs to know the exact values of those IPDs to be averaged in order to achieve a perfectly even time adjustment. In real-time communication, packets arrive and are forwarded one by one, and incoming packets should not be buffered for too long before they are sent out. This means that the watermark embedding process may need to adjust the timing of some packets and send them out before knowing the average of all the m selected IPDs. In this case, embedding the watermark bit over the average of multiple IPDs of real-time flows may lead to an uneven time adjustment over those selected packets.

V. Analysis of Probabilistic Watermarking In the Presence of Timing Perturbations

We now consider the probabilistic watermark decoding in the presence of active timing perturbation. Base on very moderate assumptions about the random timing perturbations, we first establish an upper bound of the watermark bit decoding error

probability, and then derive an approximation to the watermark bit decoding error probability.

A. Upper bound of the Watermark Bit Decoding Error Probability

Let D_i $(i = 1, \ldots, n)$ represent the random delays added to packets P_i $(i = 1, \ldots, n)$ by the adversary, let $D > 0$ be the maximum delay the adversary can add to any packet. Here we do not require the random delays added by the adversary to follow any particular distribution, except that the random delay follows some distribution of finite variance. For example, the delay distribution may be deterministic, bimodal, self-similar, or any other distribution. Furthermore, we do not require the random delay D_i's to be independent from each other, and we only assume that the covariance between different D_i's are the same.

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_{x_k} $(k = 1, \ldots, 2m)$ is independent from any random delays D_i the adversary may add. Therefore, the impact of the delays by the adversary over randomly selected P_{x_k}'s is equivalent to randomly choosing one from the random variable list $D_1, \ldots, D_n$. Let d_k $(k = 1, \ldots, 2m)$ represent the impact of the random delays by the adversary over the kth randomly selected packet P_{x_k} . Since the random delays added by the adversary follow some fixed distribution, the d_k's $(k = 1, \ldots, 2m)$ are identically distributed.

Let d_{y_k} and d_{z_k} be the random variables that denote the random delays added by the attacker to packets P_{y_k} and P_{z_k} respectively for $k = 1, \ldots, m$. Let $X_k = d_{z_k} - d_{y_k}$ be the random variable that denotes the impact of these random delays on $ipd_k = t_{z_k} - t_{y_k}$ and $\overline{X_m}$ be the random variable that denotes the overall impact of random delay on ipd_{avg}. Therefore, $E(X_k) = 0$. From equation (6), we have

$$\overline{X_m} = \frac{1}{m}\sum_{k=1}^{m}(d_{z_k} - d_{y_k}) = \frac{1}{m}\sum_{k=1}^{m} X_k \tag{7}$$

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 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 μ and σ^2 be the mean and 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.

Given $\mathrm{Cov}(u, v) = E(uv) - E(u)E(v)$, $E(d_{z_i}) = E(d_{z_j}) = E(d_{y_i}) = E(d_{y_j})$ and $\mathrm{Cov}(D_i, D_j)$ $(i \neq j)$ is constant, we have

$$E(d_{z_i}d_{z_j}) = E(d_{y_i}d_{y_j}) = E(d_{z_i}d_{y_j}) = E(d_{z_j}d_{y_i}) \tag{8}$$

Then

$$\begin{aligned} & \mathrm{Cov}(X_i, X_j) = E(X_iX_j) \\ = \ & E((d_{z_i} - d_{y_i})(d_{z_j} - d_{y_j})) \\ = \ & E(d_{z_i}d_{z_j}) + E(d_{y_i}d_{y_j}) - E(d_{z_i}d_{y_j}) - E(d_{z_j}d_{y_i}) \\ = \ & 0 \end{aligned} \tag{9}$$

Therefore

$$\begin{aligned} \mathrm{Var}(\overline{X_m}) &= \frac{1}{m^2}\mathrm{Var}(\sum_{k=1}^{m} X_k) \\ &= \frac{1}{m^2}[\sum_{k=1}^{m}\mathrm{Var}(X_k) + 2\mathrm{Cov}(X_i, X_j)] \\ &= \frac{1}{m}\mathrm{Var}(X_k) \\ &\leq \frac{4\sigma^2}{m} \end{aligned} \tag{10}$$

According to the Chebyshev inequality in statistics [7], for any random variable X with finite variance $\mathrm{Var}(X)$ and for any $t > 0$, $\Pr(|X - E(X)| \geq t) \leq \mathrm{Var}(X)/t^2$. This means that the probability that a random variable deviates from its mean by more than t is bounded by $\mathrm{Var}(X)/t^2$. By applying the Chebyshev inequality to $\overline{X_m}$ with $t = s/2$, we have

$$\Pr(|\overline{X_m}| \geq \frac{s}{2}) \leq \frac{16\sigma^2}{ms^2} \tag{11}$$

This means that the probability that the overall impact of 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 0 by increasing m, the number of redundant IPDs averaged together before embedding the watermark bit. Since the watermark bit decoding error probability is less than $\Pr(|\overline{X_m}| \geq \frac{s}{2})$, the derived upper bound is conservative and it holds true regardless of the distribution, mean or the variance of the random delays added by the attacker, or of the maximum quantization allowed for watermarking embedding. Furthermore, the upper bound of the error probability holds true even if the random delays on different packets are correlated.

B. Approximation to the Watermark Bit Robustness

In this subsection, we assume the random delays added by the adversary are independent and identically distributed (*iid*), and we derive an accurate approximation to the watermark bit robustness $\Pr(|\overline{X_m}| < s/2)$ via the well-known Central Limit Theorem of statistics [7]. Although the approximation model assumes the random delays are *iid*, our experiments (as shown in Figure 10) demonstrate that the derived approximation model can accurately model non-*iid* (e.g. batch-releasing) random delays.

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*

$$\lim_{n\to\infty}\Pr[\frac{\sqrt{n}(\overline{X_n} - \mu)}{\sigma} \leq x] = \Phi(x) \tag{12}$$

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 $\mathrm{Var}(d_{y_k}) = \mathrm{Var}(d_{z_k}) = \sigma^2$). Applying the Central Limit Theorem to random sample $X_1 = d_{z_1} - d_{y_1}, \ldots, X_m = d_{z_m} - d_{y_m}$, where $\mathrm{Var}(X_k) = \mathrm{Var}(d_{z_k}) + \mathrm{Var}(d_{y_k}) = 2\sigma^2$ and $E(X_k) = E(d_{z_k}) - E(d_{y_k}) = 0$, we have

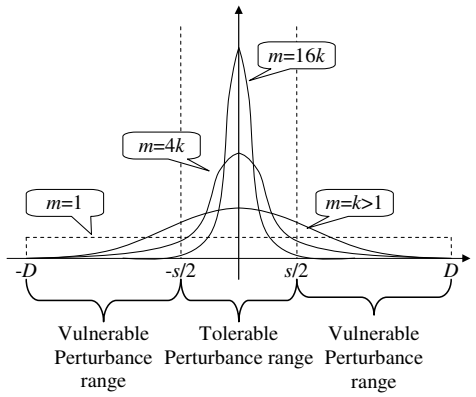

Fig. 4. Impact of Random Timing Perturbations, for Different Values of the Redundancy Number m

$$\Pr[\frac{\sqrt{m}(\overline{X_m} - E(X_i))}{\sqrt{\text{Var}(X_i)}} < x] = \Pr[\frac{\sqrt{m}\overline{X_m}}{\sqrt{2}\sigma} < x] \approx \Phi(x) \quad (13)$$

Because of the symmetry of $\Phi(x)$

$$\Pr[|\frac{\sqrt{m}\overline{X_m}}{\sqrt{2}\sigma}| < x] \approx 2\Phi(x) - 1 \quad (14)$$

Therefore,

$$p = \Pr[|\overline{X_m}| < \frac{s}{2}] = \Pr[|\frac{\sqrt{m}\overline{X_m}}{\sqrt{2}\sigma}| < \frac{s\sqrt{m}}{2\sqrt{2}\sigma}] \approx 2\Phi(\frac{s\sqrt{m}}{2\sqrt{2}\sigma}) - 1 \quad (15)$$

This means that the impact of the timing perturbation on the watermark bit is approximately normally distributed with zero mean and variance $2\sigma^2/m$. Here p represents the probability that the impact of the timing perturbation falls within range $(-\frac{s}{2}, \frac{s}{2})$. While the encoded watermark bit could be decoded correctly even when the timing perturbation falls outside $(-\frac{s}{2}, \frac{s}{2})$, such a probability is small when p is close to 1. In the rest of this paper, we will use p as a conservative approximation to the probability that the watermark bit will survive the timing perturbation.

Equation (15) confirms the result of (11). Figure 4 illustrates how the distribution of the impact of random timing perturbations by the attacker can be "squeezed" into the tolerable perturbation range by increasing the number of redundant IPDs averaged.

Equation (15) also gives us an accurate estimate of the watermark bit robustness. For example, assume the maximum delay by the attacker is normalized to 1 time unit, the random delays added by the attacker are uniformly distributed over [0, 1] (whose variance σ^2 is 1/12), $s = 0.4$, and $m = 12$, then $\Pr[|\overline{X_{12}}| < 0.2] \approx 2\Phi(1.2 \times \sqrt{2}) - 1 \approx 91\%$. In other words, the impact of random timing perturbations on the average of 12 IPDs, with about 91% probability, will fall within the range [-0.2, 0.2]. Table I shows the estimation and simulation results of watermark bit robustness with uniformly distributed random delays over [0, 1], $s = 0.4$ and various values for m. This demonstrates that the Central Limit Theorem can give us a very accurate estimate for a sample size as small as $m = 7$.

From equation (15), it can be seen that it is easier to achieve a target level of robustness by increasing s than by increasing m. For example, the effect of increasing s by a factor of 2 is the same as that of increasing m by a factor of 4.

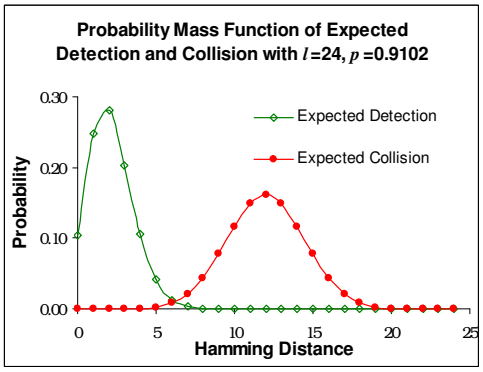

Fig. 5. Effect of the Threshold h on the Detection and Collision Rates of the Watermarking Method

C. Analysis of Watermark Detectability

Watermark detection refers to the process of determining if a given watermark is embedded into the IPDs of a specific connection or flow. Let the secret information shared between the watermark embedder and decoder be represented as $< S, m, l, s, w >$, where S is the packet 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 w is the l-bit watermark to be detected. 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 robustness of the watermark detector against timing perturbations by the attacker. Given any quantization step size s, there is a non-zero probability that an embedded watermark bit will be corrupted by timing perturbations, no matter how much redundancy is 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 w_f, the expected watermark detection rate will be

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

For example, for the value $p = 0.9102, l = 24, h = 5$, 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 that an unwatermarked flow happens to have the watermark to be detected naturally. In this case, the watermark detector would report the unwatermarked flow as having the watermark. It is termed a *collision* between w and f if

m	7	8	9	10	11	12
Estimated Robustness (%)	80.46	83.32	85.54	87.86	89.58	91.02
Simulated Robustness (%)	80.27	83.27	85.68	87.79	89.54	91.02

TABLE I
WATERMARK BIT ROBUSTNESS SIMULATION FOR UNIFORMLY DISTRIBUTED RANDOM DELAYS OVER [0, 1], $s = 0.4$

$H(w_f, w) \leq h$ for an unwatermarked flow f. Collisions are obviously undesirable, as they may lead to false conclusions about who the source of an attack is.

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

$$\sum_{i=0}^{h} \left(\begin{array}{c} l \\ i \end{array} \right) (\frac{1}{2})^l \tag{17}$$

Figure 5 shows the derived probability distribution of the expected watermark detection and collision rates for $l = 24$ 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 h would be the one that gives high expected detection rate, while keeping the false positive rate low. However, the optimal h and l depend on 1) the number of packets available; 2) the defining characteristics of the timing perturbation; and 3) the desired level of effectiveness. Due to space limitation, we leave the derivation of optimal h and l as a future work. We show that our flow watermarking scheme can be effective even with potentially suboptimal h and l.

Given any quantization step size $s > 0$, and 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 $0 \leq h < l$. Assuming that h is chosen such that $h < l/2$, then we have

$$\sum_{i=0}^{h} \left(\begin{array}{c} l \\ i \end{array} \right) (\frac{1}{2})^l \quad \leq \quad \sum_{i=0}^{h} \left(\begin{array}{c} l \\ h \end{array} \right) (\frac{1}{2})^l \leq (h+1)\frac{l^h}{2^l} \tag{18}$$

Because $\lim_{l \to \infty} \frac{l^h}{2^l} = 0$, we can always make the expected watermark collision probability $\sum_{i=0}^{h} \left(\begin{array}{c} l \\ i \end{array} \right) (\frac{1}{2})^l < P_c$ by having sufficiently large watermark bit number l. Since $\sum_{i=0}^{h} \left(\begin{array}{c} l \\ i \end{array} \right) p^{l-i}(1-p)^i \geq p^l$, we can always make the expected detection rate $\sum_{i=0}^{h} \left(\begin{array}{c} l \\ i \end{array} \right) p^{l-i}(1-p)^i > P_d$ by making p sufficiently close to 1. From inequality (11), this can be accomplished by increasing the redundancy number m given any fixed values of s and σ.

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

In practice, the number of packets available is the fundamental limiting factor to the achievable effectiveness of our watermark based correlation. Our experiences show that our watermark based correlation can be effective with as few as several hundred packets. For example, the experiments in sections VII-A and VII-B show that the watermarking only requires less than 300 packets to achieve a virtually 100% decoding rate against up to 1000ms random timing perturbation, and less than 0.35% false positive rate.

Although the watermark detection true positive rate and false positive rate can be made arbitrarily close to 100% and 0% at the same time by introducing enough redundancy, there is always a non-zero probability that an embedded watermark is not detected. In fact, there exist some special case of timing perturbation that could potentially completely remove the embedded watermark. For example, with sufficiently large delay, the timing of the watermarked packet flow could be perturbed such that the inter-packet arrival time is constant (or equivalently, the IPDs between adjacent packets equal to the average IPDs). In this case the watermark decoding of the perturbed flow would be fixed no matter what and how the watermark has been embedded. However, achieving such a complete elimination of embedded watermark requires knowing the exact timing characteristics of the traffic in advance. In the following section we analyze the negative impacts of the adversary's timing perturbations, and their limits under the constraints of real-time communication.

VI. LIMITS OF ADVERSARY'S TIMING PERTURBATION

We have shown that there exist special cases of brute force timing perturbation that could completely remove any embedded watermark from any distribution of inter-packet timing.

In this section, we analyze the limitations on the negative impact of the adversary's timing perturbations. We assume that the key parameters of the watermark embedding method are unknown to the adversary. We first identify the minimum distortion required for the adversary to completely remove the embedded watermark and the optimal strategy for doing so. We then analyze the additional constraints imposed by real-time communication and their implications for the adversary's ability to interfere with or distort the watermark. We show that it is generally infeasible for the adversary to completely eliminate the embedded watermark from a flow of packets in real-time.

A. Minimum Brute Force Perturbation Needed to Completely Remove Watermark

Let the n-dimension vector $S^N = < S_1, \ldots, S_N >$ (where $S_i \in R^+$ is a random variable) be the host signal or carrier in which the watermark is to be embedded, M be the message to be embedded and transferred, and K be the key information for correct information embedding and decoding. Let $X^N = < X_1, \ldots, X_N >$ (where $X_i \in R^+$ is a random variable) be the signal after M is embedded in S^N. The adversary distorts X^N into another vector $Y^N = < Y_1, \ldots, Y_N >$ (where $Y_i \in R^+$ is

another random variable). We can view Y_i as an estimator of X_i and use the *mean squared error* $MSE(X_i, Y_i) = E[(X_i - Y_i)^2]$ to measure the distortion between random variables X_i and Y_i. We use $D(X^N, Y^N) = \frac{1}{N}\sum_{i=1}^{N} MSE(X_i, Y_i)$ to measure the overall distortion between X^N and Y^N.

We first consider the distortion between single random variable X_i and Y_i. From an information-theoretic point of view, to eliminate all the hidden information in X_i from Y_i via brute force is to make the mutual information between X_i and Y_i $I(X_i; Y_i)$ be 0. That is

$$I(X_i; Y_i) = H(X_i) - H(X_i|Y_i) = 0 \tag{19}$$

or

$$H(X_i) = H(X_i|Y_i) = H(X_i, Y_i) - H(Y_i) \tag{20}$$

Then we have

$$H(X_i, Y_i) = H(X_i) + H(Y_i) \tag{21}$$

Therefore, X_i and Y_i are independent from each other. This means that the adversary needs to distort X_i into another independent random variable Y_i in order to completely remove any hidden information from random variable X_i via brute force. The following theorem[1] determines the MSE to achieve this.

Theorem 3: The mean square error between two independent random variables X and Y is

$$MSE(X, Y) = Var(X) + Var(Y) + ((E(X) - E(Y))^2$$

Both $Var(Y)$ and $(E(X) - E(Y))^2$ are non-negative, and they will be 0 only when Y equals the constant value $E(X)$.

Corollary 1: The minimum mean square error needed to convert one random variable X into another independent random variable Y is $Var(X)$, and it only occurs when Y equals to constant value $E(X)$.

Therefore, the minimum distortion required to completely eliminate hidden information from any particular random variable X via brute force is $Var(X)$, and the optimal strategy to do so is to convert X into constant value $E(X)$.

Now we consider the distortion between two n-dimensional vectors X^N and Y^N. The overall distortion between X^N and Y^N ($D(X^N, Y^N)$) will reach its minimum when each $MSE(X_i, Y_i)$ reaches its minimum. Therefore, the minimum overall distortion $D(X^N, Y^N)$ required to completely eliminate hidden information from X^N is $\frac{1}{N}\sum_{i=1}^{N} Var(X_i)$, and the optimal strategy to achieve this is to convert $X^N = < X_1, \ldots, X_N >$ into $Y^N = < E(X_1), \ldots, E(X_N) >$. Given any fixed $E(X_1), \ldots, E(X_N)$, Y^N is fixed regardless of the exact values of X^N. Therefore, the mutual information between X^N and Y^N is zero in this case. This result holds true regardless of the distribution of each X_i in X^N.

This result is consistent with Moulin's work [16], [15] regarding the achievable capacity of an information hiding scheme in the presence of distortion by an adversary. Moulin showed that for a normally distributed host signal, the information hiding capacity is 0 when the distortion by the adversary is equal to or greater than the variance of the host signal. Here we have shown that the adversary could remove any hidden information from the host signal of any distribution via distortion equal to or greater than the variance of the host signal, and we have described an optimal strategy to eliminate any hidden information via brute force distortion.

[1]We omit the proof due to space limitation

B. Constraints of Real-Time Communication and Their Implications

This section considers the additional constraints imposed by real-time communication and their implications for the adversary's capability to completely remove any hidden information from a real-time packet flow.

For a real-time packet flow $P_1, \ldots, P_n$ with time stamps $t_1, \ldots, t_n$ respectively, let the host signal in which information will be embedded be $S^N = < t_1, \ldots, t_n >$. The requirements of real-time communication impose the following constraints on any distortions over the packet timing.

1) Each packet can only be delayed.
2) The delay to any packet is bounded (finite), otherwise the real-time communication is broken.
3) The delay to packet P_k ($k < n$) has to be determined and performed before all n packets are received, otherwise the delay to P_k is unbounded when $n \to \infty$.

Let δ_i be the delay added to packet P_i, and t'_i be the distorted time stamp of packet P_i, then $t'_i = t_i + \delta_i$. The original and distorted inter-packet delays (IPD) between P_{i+1} and P_i are $I_i = t_{i+1} - t_i$ and $I'_i = t'_{i+1} - t'_i$ respectively. Therefore,

$$\delta_k = t'_k - t_k = \delta_1 + \sum_{i=1}^{k-1} (I'_i - I_i) \tag{22}$$

The original and the perturbed inter-packet timing characteristics of packet flow $P_1, \ldots, P_n$ can be represented by $< t_1, I_1, \ldots, I_{n-1} >$ and $< t'_1, I'_1, \ldots, I'_{n-1} >$ respectively. In particular, $< I'_1, \ldots, I'_{n-1} >$ represents the distortion pattern over the original inter-packet timing characteristics.

According to results from section VI-A, in order to completely remove any hidden information from the original inter-packet timing characteristics, the adversary needs to disturb $< t_1, I_1, \ldots, I_{n-1} >$ into an independent one. That means $< I'_1, \ldots, I'_{n-1} >$ needs to be independent from $< I_1, \ldots, I_{n-1} >$. Therefore, the distortion pattern $< I'_1, \ldots, I'_{n-1} >$ can be thought to be pre-determined before the original inter-packet timing characteristics $< t_1, I_1, \ldots, I_{n-1} >$ is ever known.

Let I_{min} and I_{max} denote the minimum and the maximum of all I_i's respectively and let $D > 0$ represent the arbitrarily large maximum delay that the adversary could add to any packet. To satisfy the real-time constraints, the adversary could buffer at most $\frac{D}{I_{min}}$ packets before the delay δ_1 is determined and applied to packet P_1.

Assume the adversary buffers b ($0 < b < n$) packets: $P_1, \ldots, P_b$ before decides δ_1. Since the adversary knows $< t_1, I_1, \ldots, I_{b-1} >$ and all I'_k, he can find appropriate value of δ_1 to make sure that $\delta_1, \ldots, \delta_b$ are non-negative according to equation (22).

Now we show that when $I_{min} < I_{max}$, it is generally infeasible for the adversary to bound δ_n within range $[0, D]$ without prior knowledge of the exact values of $I_b, \ldots, I_{n-1}$. From equation (22), it is easy to get

$$\delta_n = \delta_b + \sum_{i=b}^{n-1} (I'_i - I_i) \tag{23}$$

Therefore, the real-time constraint $x_n \in [0, D]$ is equivalent to

$$-\frac{\delta_b}{n} \le \frac{1}{n}\sum_{i=b}^{n-1} I'_i - \frac{1}{n}\sum_{i=b}^{n-1} I_i \le \frac{D - \delta_b}{n} \tag{24}$$

Since both δ_b and D are fixed and finite, we have

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i\,=\,b}^{n-1} I'_i = \lim_{n\to\infty}\frac{1}{n}\sum_{i\,=\,b}^{n-1} I_i \tag{25}$$

This means that the average inter-arrival time (or equivalently the average packet rate) of the perturbed packet flow must be arbitrarily close to that of the original packet flow given sufficiently large n. In order to completely remove any hidden information from the inter-packet timing domain of a packet flow, $< I'_b, \ldots, I'_{n-1} >$ must be independent from $< I_b, \ldots, I_{n-1} >$. That means $< I'_b, \ldots, I'_{n-1} >$ must be determined before $< I_b, \ldots, I_{n-1} >$ is ever known. However, when $I_{min} < I_{max}$ and n is large, it is infeasible for the adversary to determine $< I'_b, \ldots, I'_{n-1} >$ whose average inter-arrival time is arbitrarily close to an unknown value $\frac{1}{n}\sum_{i\,=\,b}^{n-1} I_i \in [I_{min}, I_{max}]$.

In other words, when $I_{max} > I_{min}$, the adversary is not able to meet all the real-time constraints when he tries to completely remove all the hidden information from a sufficiently long real-time packet flow. Therefore, it is generally infeasible for the adversary to completely eliminate all the hidden information in real-time from a sufficiently long packet flow even with arbitrarily large delays.

VII. Implementation and Experiments

We have implemented both online and offline versions of our watermarking embedding and decoding scheme. The online version runs as a Linux kernel module in Linux kernel 2.4.22, and it embeds the specified watermark to specified IP flow at real-time. The online version uses Linux netfilter and iptable to communicate the watermarking parameters from user space to kernel space and it has 10ms precision in adjusting the timing of selected packets. The offline version is a user space application that reads and manipulates the flow trace files rather than real-time flows. The watermark embedding part reads a flow trace file of pcap format, and outputs a new flow trace file in which the watermark is embedded in the packet timing. The watermark detecting part reads the watermarked flow trace file and reports if the specified watermark is detected. Our experience has shown that the online version and offline version of our implementations consistently give almost identical results on correlation true positive and false positive rates.

In this section, we empirically validate our active watermark based correlation scheme in the presence of random timing perturbations. In particular, we seek to answer the following questions:

1) How vulnerable are existing (passive) timing-based correlation schemes to random timing perturbations?
2) How robust is the active watermark-based correlation against random timing perturbation?
3) How effective is watermark-based correlation in correlating the encrypted flows in the presence of both *iid* and non-*iid* random timing perturbations?
4) How accurate are our quantitative tradeoff models of watermark bit robustness, watermark detection rate and watermark collision rate in predicting the actual values?

We have used three flow sets, labelled FS1, FS1-Int and FS2, in our experiments. FS1 is derived from over 49 million packet headers of the Bell Labs-1 traces of NLANR [17]. It contains 121 SSH flows that have at least 600 packets and that are at least 300 seconds long. FS2 contains 1000 synthetic telnet flows generated from an empirically-derived distribution [6] of telnet packet inter-arrival times, using the tcplib [5] tool.

We also wish to examine the performance of our method for purely interactive traffic. Because SSH flows may contain non-interactive traffic such as those of bulk data transfer and X-windows management, it is desirable to remove such non-interactive traffic from the SSH flows to get more accurate evaluation of watermark-based correlation of interactive flows. We examine the adjacent IPD of those SSH flows in FS1, and filter out those flows whose adjacent IPDs are too short to be generated by humans. Since it is very unlikely for a person to type more than 14 keystrokes per second, we chose 70ms as the threshold to tell whether the adjacent IPD is generated by human typing or not. We have found 33 out of 121 flows in FS1 satisfy the following conditions:

1) Flow has $> 40\%$ adjacent IPDs shorter than 70ms.
2) Flow has $> 10\%$ 10-consecutive adjacent IPDs all of which are shorter than 70ms.

By removing 33 non-interactive flows from FS1, we obtained a new flow set FS1-Int following the above definition.

We considered three types of timing perturbations in our experiments. The first is the *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 a *self-similar perturbation*, in which the attacker at a stepping stone adds to each packet a self-similar random delay. The third is the *batch-releasing perturbation*, in which the attacker at a stepping stone periodically buffers and holds all packets received within a certain time window, and then forwards all the buffered packets at line speed once the time window has expired.

The first type of perturbation is *iid*, while the second and third types of perturbation are non-*iid*. In fact, the batch-releasing perturbation is neither independent nor identically distributed, since the impact over any packet is closely correlated to the time of that packet. In addition, batch-releasing drastically changes the original timing characteristics of any flow to a pattern of periodic bursts (as shown in Figure 6), which represents a challenging case for any timing based correlation.

A. Correlation True Positive Experiments

This set of experiments aim to compare and evaluate the correlation effectiveness of our proposed active watermark based correlation and previous passive timing-based correlation under various timing perturbations. We used two methods of correlation. First, we used an existing, *passive* timing-based correlation method called IPD-based correlation [32] to correlate each flow in FS1 with the same flow, after it is perturbed by various levels of uniformly distributed random timing perturbations. If the flow and the perturbed flow are reported correlated, it is considered as a *true positive* ("IPDCorr 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 requires 300 packets to be selected, of which 288 were delayed. Figure 7 shows the effect of the watermark embedding over the inter-packet timing, and illustrates that the watermark embedding is far from obvious. Third, we randomly perturbed the packet timing of the watermarked flows of FS1 and FS2 with different types of timing

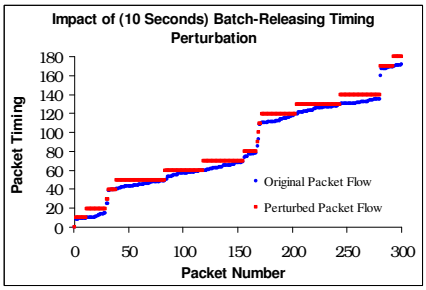

Fig. 6. Effect of 10 Seconds Batch-Releasing Timing Perturbation

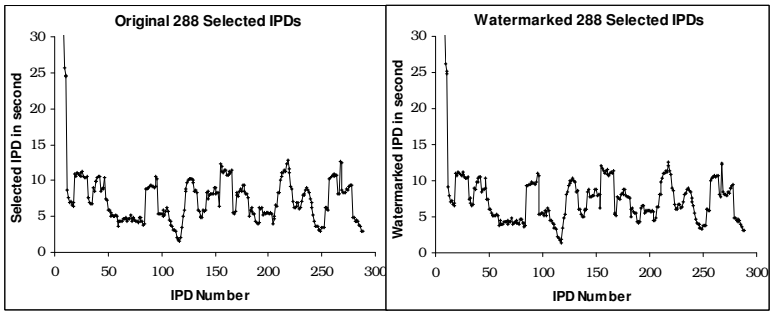

Fig. 7. Comparison of 288 Selected IPDs Before and After Watermark Embedding

perturbations. It is considered a *true positive* ("IPDWMCorr TP") 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 watermark detection rate from equations (15) and (16) under various maximum delays of uniformly distributed random timing perturbations.

1) Correlation under Uniformly Distributed Random Timing Perturbation: Figure 8 shows the measured (as well as expected) true positive rates of IPD-based correlation and watermark-based correlation on FS1 and FS2 under various levels of uniformly distributed random timing perturbations. The results clearly indicate that IPD-based correlation is vulnerable to even moderate random timing perturbations. Without timing perturbation, IPD-based correlation is able to successfully correlate 93% 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 with a 200ms maximum delay, the rate drops to 21.5%.

In contrast, the proposed watermark-based correlation of the flows in FS1, FS1-Int and FS2 is able to achieve virtually a 100% true positive rate, with up to a maximum 600ms random timing perturbation. With a maximum 1000ms timing perturbation, the true positive rates of watermark-based correlation for FS1, FS1-Int and FS2 are 84.2%, 89.85% 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 (15) and (16). In particular, the true positive rate measurements of FS2 are very close to the predicted values at all perturbation levels.

2) Correlation under Self-Similar Distributed Random Timing Perturbation: To see how our watermark-based correlation works when the random timing perturbation is non-*iid*, we first investigated correlation under self-similar timing perturbations. We used Glen Kramers implementation [13] of Taqqu et als [29] self-similar synthetic traffic generating method to generate the self-similar timing perturbation with 128 sources of ON/OFF periods aggregated with 30% cumulative load. And we have bounded the timing perturbation through modulo operation. In particular, we have used 128 aggregating sources of ON/OFF periods to generate self-similar delays in units of milliseconds.

Figure 9 shows the measured watermark correlation true positive rates under various bounded self-similar timing perturbation, and expected watermark correlation true positive rates of uniformly distributed random delays, with h=5; l=24; m=12; s=400ms. It shows that the bounded self-similar perturbation yields much higher watermark correlation true positive rates than the expected true positive rates of uniformly distributed random delay perturbation with the same delay upper bound (1200ms $\sim$ 2400ms). This indicates that the bounded self-similar timing perturbation has less negative impact than the uniformly distributed random timing perturbation of same upper bound.

We calculated the variance of 10000 self-similar delays that are bounded by 1000ms, and obtained $\sigma^2_{selfsim} = 46786$. However, the variance of a uniformly distributed random variable of range [0, 1000] $\sigma^2_{uniform}$ is 83333. According to equation 15, the smaller the variance σ^2 is, the higher is the watermark bit robustness (equivalently, the higher the watermark true positive rate should be). This explains why the self-similar type of random perturbation has less negative impact than the uniformly distributed random delays of same upper bound.

3) Correlation under Batch-Releasing Random Timing Perturbation: The second type of non-*iid* random timing perturbation we investigated is the batch-release timing perturbation. In this model, incoming packets are buffered until expiration of the next timer period, at which point all buffered packets are output at line speed, in a burst. With the batch-release perturbation, the actual delay of any packet depends on where it falls within the timer interval, or window. Assuming the packet arrival times are uniformly distributed within the batch release window, we are able to calculate the variance of the delays over all packets, given the window size or duration.

Figure 10 shows the measured and estimated correlation true positive rates of our watermark-based correlation on flow set FS2, under various levels of batch-release timing perturbations. We used 24-bit watermarks with quantization step size s=400ms, redundancy number m=12, and Hamming distance threshold h=5. The expected true positive rates are calculated by equations (15) and (16). The measured values are close to the expected values, which demonstrates that our analytical model is applicable to non-*iid* timing perturbations.

B. Correlation False Positive Experiments

As mentioned above, there is a non-zero probability that an unwatermarked flow happens to exhibit the randomly chosen watermark. This case is considered a correlation collision, or false positive. According to our correlation collision model (17), the collision rate is determined by the number of watermark bits l and the Hamming distance threshold h.

We experimentally investigated the following false positive rates for varying values of the Hamming distance threshold h and the number of watermark bits l:

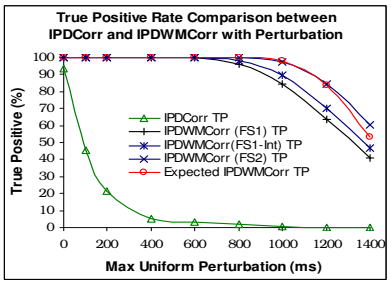

Fig. 8. Correlation True Positive Rates under Uniformly Distributed Random Timing Perturbations

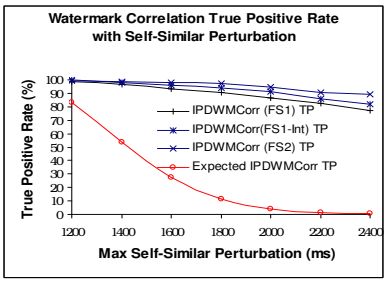

Fig. 9. Correlation True Positive Rates under Self-Similar Random Timing Perturbations

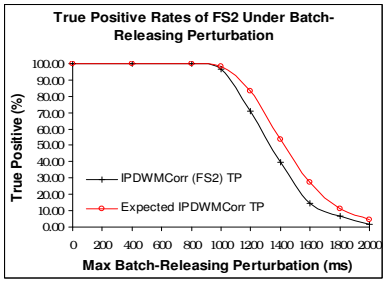

Fig. 10. Correlation True Positive Rates under Batch-Releasing Random Timing Perturbations

1) Collision rates between a given flow and 10,000 1,000,000 randomly generated 24-bit watermarks with varying Hamming distance threshold h.
2) Collision rates between a given 24-bit watermark and 10,000 1,000,000 randomly generated (using tcplib) synthetic telnet flows with varying Hamming distance threshold h.
3) Collision rates between a given flow and 100,000 randomly generated watermarks of various lengths with Hamming distance threshold h=5.
4) Collision rates between a given watermark of various lengths and 10,000 1,000,000 randomly generated (using tcplib) synthetic telnet flows with Hamming distance threshold h=5.

Figure 11 shows the measured and estimated correlation false positive rates. The left sub-figure shows the false positive rates for varying Hamming distance thresholds and fixed length (24-bit) watermarks, and the right sub-figure shows the false positive rates for varying watermark lengths and a fixed Hamming distance threshold h=5. The measured values are the average of 100 separate experiments and they are very close to the estimated values calculated from equation (17). Thus the experimental results validate our model of the collision probability.

C. Watermark Detection Tradeoff Experiments

Equation (15) gives us the quantitative tradeoff between the expected watermark bit robustness, the redundancy number m and the defining characteristics of the random timing perturbation σ^2. With a given watermark bit robustness p, equation (16) gives us the expected watermark detection rate.

To verify the validity and accuracy of our tradeoff models of watermark bit robustness and watermark detection rate, we did the following experiments:

1) We embedded a random 24-bit watermark into each flow in FS1 FS1-Int and FS2, with quantization step s=400ms, and varying redundancy numbers $m = 7, 8, 9, 10, 11, 12$. After we perturbed the watermarked flows with 1000ms maximum random delays, we measured the watermark detection rate of the perturbed, watermarked flows with Hamming distance threshold h=5.
2) We also embedded a random 24-bit watermark into each flow in FS1, FS1-Int and FS2, with quantization step s=400ms, redundancy number m=12. After perturbing the watermarked flows with 1000ms maximum random delays, we measured the watermark detection rate of the perturbed, watermarked flows for varying Hamming distance thresholds of $h = 2, 3, 4, 5, 6, 7, 8$.
3) In a separate experiment, we embedded a random watermark of varying lengths $l = 18, 19, 20, 21, 22, 23, 24$ into each flow in FS1, FS1-Int and FS2, with quantization step s=400ms and redundancy number m=12. After perturbing the watermarked flows with 1000ms maximum uniform random delays, we measured the watermark detection rate of the perturbed, watermarked flows with different Hamming distance threshold h=3.

Figure 12 shows the average of 100 experiments for the measured watermark detection rates of FS1 and FS1-Int, and the average of 10 experiments for the measured watermark detection rates of FS2, as well as the expected detection rate derived from equations (15) and (16). In all cases, the measured detection rates of FS2 are almost identical to the expected values, and detection rates of FS1 are similar to but lower than the expected values. The detection rates of FS1-Int are always between that of FS1 and FS2. These results validate our quantitative tradeoff models of watermark bit robustness and watermark detection rate.

D. Comparison with Other Representative Passive Approach

We have also compared the effectiveness and the numbers of packets needed by our active watermark correlation approach, and a representative passive correlation approach [1], under identical levels of timing perturbation on the same sets of traces.

For Poisson arrivals, paper [1] claimed that its detection algorithm `DETECT-ATTACKS(`δ`,` p_Δ`)` is guaranteed to achieve a 100% detection rate and no more than δ false positive rate given sufficient number of packets. However, it did not include any experimental results that demonstrated the claimed effectiveness. To empirically compare the effectiveness of our active approach and that of the passive correlation method of [1], we implemented its detection algorithm `DETECT-ATTACKS(`δ`,` p_Δ`)`. After identifying the maximum number of packets p_Δ in any time interval Δ=800ms from each of the 1000 flows in FS2, we applied detection algorithm `DETECT-ATTACKS(`δ`,` p_Δ`)`, with δ=0.3%, to correlate flows in FS2 and the corresponding perturbed flows with maximum 800ms uniformly distributed perturbation. Surprisingly, the detection algorithm `DETECT-ATTACKS(`δ`,` p_Δ`)` in this test only achieved a 79.5% detection rate, while experiencing no false positives.

As shown in section VII-A, under a maximum 800ms uniformly distributed timing perturbation, our watermark-based correlation method achieved at least a 99.9% true positive rate and about a 0.3% false positive rate, using parameter values of h=5, l=24, s=400ms and m=12 on flow set FS2.

We have further compared the upper bounds of the number of packets needed by our active correlation approach, and by

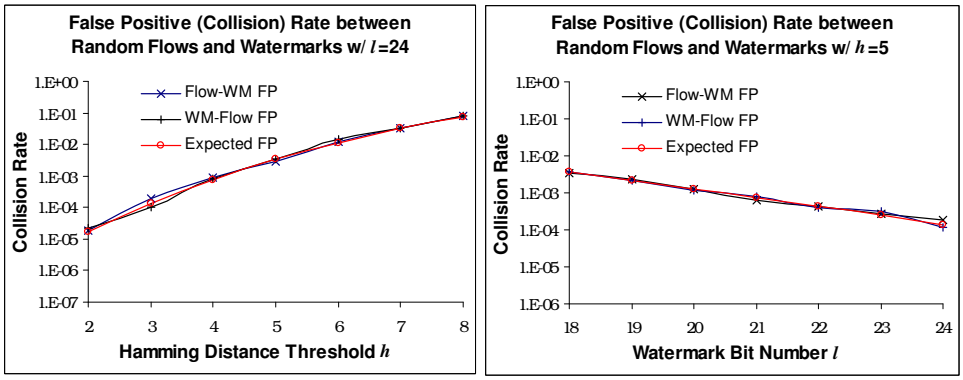

Fig. 11. Correlation False Positive (Collision) Rates vs Hamming Distance Threshold h and Number of Watermark Bits l

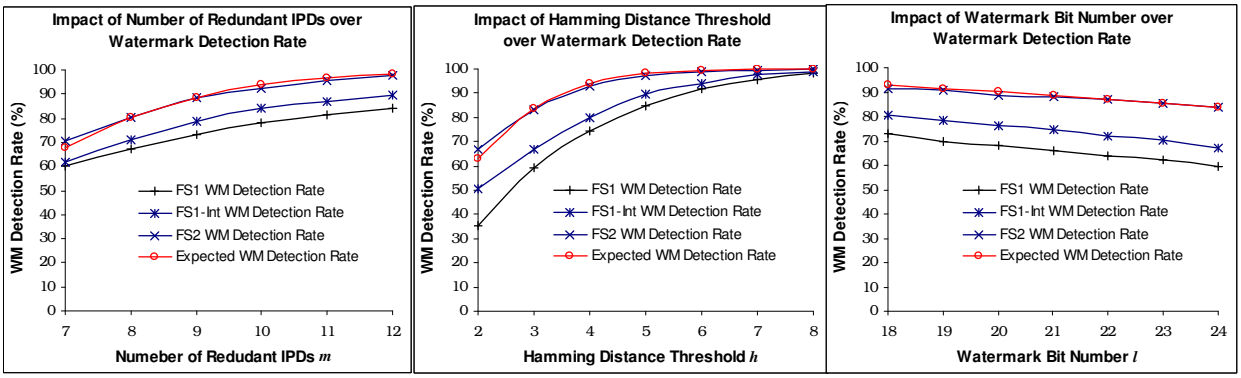

Fig. 12. Watermark Detection Rate Tradeoff with Redundancy Number m, Hamming Distance Threshold h and Number of Watermark Bits l

the method of [1]. As an example, consider the requirements to achieve a guaranteed correlation effectiveness of at least a 99.9% true positive rate (TPR) and a 0.35% false positive rate (FPR), for a maximum 600ms uniformly distributed timing perturbation. Given the parameter values m=36, s=400ms, and D=600ms, inequality (11) guarantees that the upper bound of the bit error probability of our active approach will be less than 0.0834 (i.e., $p > 0.9166$). Choosing l=32, h=8, and p=0.9166, the expected watermark detection rate from equation (16) is >99.9%, and the expected watermark collision rate from equation (17) is 0.35%. Therefore, in order to achieve a 99.9% TPR and a 0.35% FPR with a maximum 600ms timing perturbation, our active approach needs to adjust the timing of no more than 32×36 = 1,152 packets. For the approach of [1], letting $\Delta = 600$ms, $\delta = 0.35\%$ and using the value of p_Δ measured from the flows in FS1, detection algorithm DETECT-ATTACKS(δ, p_Δ) requires at most 16,698 packets to achieve a 100% TPR and a 0.35% FPR. For this target, the upper bound of the number of packets needed by the passive approach of [1] to achieve a comparable correlation effectiveness is at least an order of magnitude more than that of the active approach.

VIII. Conclusion and Future Works

Tracing attackers' traffic through stepping stones is a challenging problem, especially when the attack traffic is encrypted, and its timing is manipulated (perturbed) to interfere with traffic analysis. The random timing perturbation by the adversary can greatly reduce the effectiveness of passive, timing-based correlation techniques.

We presented a novel active timing-based correlation approach to deal with random timing perturbations. By embedding a unique watermark into the inter-packet timing, with sufficient redundancy, we can make the correlation of encrypted flows substantially more robust against random timing perturbations. Our analysis and our experimental results confirm these assertions.

Our watermark-based correlation is provably effective against correlated random timing perturbation as long as the covariance of the timing perturbations on different packets is fixed. Specifically, *the proposed watermark-based correlation can, with arbitrarily small average time adjustment, achieve arbitrarily close to 100% watermark detection (correlation true positive) rate and arbitrarily close to 0% collision (correlation false positive) probability at the same time against arbitrarily large (but bounded) random timing perturbation of arbitrary distribution (or process), as long as there are enough packets in the flow to be watermarked.*

Compared with previous passive correlation approaches, our active watermark-based correlation has several advantages:

- Our active watermark-based correlation makes no assumptions about the original distribution of the inter-packet timing of the original packet flow, and it does not require the adversary's timing perturbation to follow any specific distribution or random process to be effective. This is in contrast to existing passive timing based correlation methods, all of which assumed the inter-packet timing of the original packet flow to follow some specific distribution or random process (e.g. Poisson) when deriving their bounds.
- Our active watermark-based correlation was shown to require substantially fewer packets than a representative passive timing-based correlation method to achieve a given level of robustness.

- The effectiveness of our active watermark-based correlation can be modelled more accurately. Besides identifying the provable upper bounds on the number of packets needed to achieve desired correlation effectiveness under any given level of perturbation, we have also identified the quantitative tradeoff models between the number of packets needed to achieve any desired correlation effectiveness under any given level of perturbation. 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.

We have experimentally investigated the watermark-based correlation under both *iid* and non-*iid* timing perturbations, and the experimental results confirmed our analytical conclusion that our watermark-based correlation is effective for both *iid* and non-*iid* random timing perturbations.

While our flow watermarking approach has been shown to be promising in correlating encrypted traffic in the presence of timing perturbation, it is not optimal from coding theoretic perspective. One interesting area of future work is to investigate how to make the flow watermarking more robust with fewer packets.

ACKNOWLEDGMENT

This work was partially supported by NSF Grants CNS-0524286 and CCF-0728771. Some preliminary results of this work have been presented in the *10th ACM Conference on Computer and Communication Security (CCS 2003)*, October, 2003 [31].

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] R. C. Chakinala, A. Kumarasubramanian, R. Manokaran, G. Noubir, C. Pandu Rangan, and R. Sundaram. Steganographic Communication in Ordered Channels. In *Proceedings of the 8th Information Hiding International Conference (IH 2006)*, 2006.

[3] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

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

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

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

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

[8] 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.

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

[10] T. He and L. Tong, Detecting Encrypted Stepping-Stone Connections. In *IEEE Transactions on Signal Processing*, 55(5), pages 1612-1623, 2006.

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

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

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

[14] 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.

[15] P. Moulin. Information-Hiding Games. In *Proceedings of International Workshop on Digital Watermarking (IWDW 2003)*, LNCS–2613, May 2003.

[16] P. Moulin and J.A. O'Sullivan. Information-Theoretic Analysis of Information Hiding. In *IEEE Transaction on Information Theory*, 49(3), pages 563–593, March 2003.

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

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

[19] P. Peng, P. Ning, D. S. Reeves, On the Secrecy of Timing-Based Active Watermarking Trace-Back Techniques. In *Proceedings of the 2006 IEEE Symposium on Security & Privacy (S&P 2006)*, May 2006.

[20] P. Peng, P. Ning, D. Reeves and X. Wang. Active Timing-Based Correlation of Perturbed Traffic Flows with Chaff Packets. In *Proceedings of the 2nd International Workshop on Security in Distributed Computing Systems (SDCS-2005)*, June, 2005.

[21] Y. J. Pyun, Y. H. Park, X. Wang, D. S. Reeves and P. Ning. Tracing Traffic through Intermediate Hosts that Repacketize Flows. In *Proceedings of the 26th Annual IEEE Conference on Computer Communications (Infocom 2007)*. May 2007.

[22] Y.J. Pyun and D. S. Reeves. Deployment of Network Monitors for Attack Attribution. To appear in *Proceedings of the Fourth International Conference on Broadband Communications, Networks, and Systems (IEEE Broadnets 2007)*, September 2007.

[23] 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.

[24] C. E. Shannon. A Mathematical Theory of Communication. In *Bell System Technical Journal*, 27, pages 379–423 and 623-656, July and October, 1948.

[25] 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.

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

[27] 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.

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

[29] 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.

[30] X. Wang, S. Chen and S. Jajodia. Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems. In *Proceedings of the 2007 IEEE Symposium on Security & Privacy (S&P 2007)*, May 2007.

[31] 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 Communications Security (CCS 2003)*, pages 20–29. ACM, October 2003.

[32] X. Wang, D. Reeves, and S. F. 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.

[33] X. Wang, D. Reeves, S. F. 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.

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

[35] 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.

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

[37] L. Zhang, A. G. Persaud, A. Johnson, and Y. Guan. Detection of Stepping Stone Attack under Delay and Chaff Perturbations. In *Proceedings of the 25th IEEE International Performance Computing and Communications Conference (IPCCC 2006)*, April 2006.

Xinyuan Wang is an Assistant Professor of Computer Science at George Mason University. He received his Ph.D. in Computer Science from North Carolina State University in 2004 after years professional experience in networking industry. His main research interests are around computer network and system security including malware analysis and defense, attack attribution, anonymity and privacy, VoIP security, digital forensics. Xinyuan Wang is a recipient of the NSF Faculty Early Career Development (CAREER) Award.

Douglas Reeves received his Ph.D. in Computer Science from the Pennsylvania State University in 1987. In the same year he joined the faculty of N.C. State University, where he is currently Professor of Computer Science. His research interests include computer systems, security, and peer-to-peer computing.