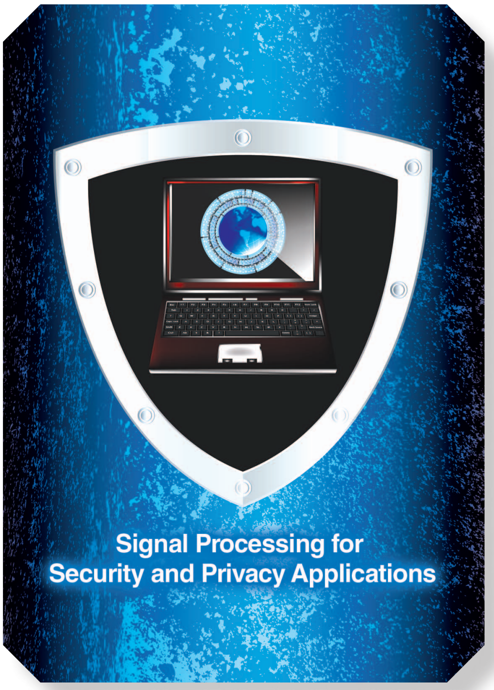
Sachin Kadloor, Parv Venkitasubramaniam, and Negar Kiyavash

Preventing Timing Analysis in Networks

A statistical inference perspective

Information security requires the protection of not only the contents of data transmitted but also the timing of network operations. Knowledge of timing alone can reveal to an adversary the nature of users' online behavior, including, but not restricted to, Web sites accessed, recipients of e-mails and chat messages, financial resources considered, and more. It is imperative that current network protocols be redesigned to prevent the retrieval of packet timing in a network and limit the "networking" information inferable from the timing. Timing information can be retrieved by adversaries both passive and active: active adversaries masquerade as network users and retrieve the timing of other legitimate users using delays experienced at shared schedulers, while passive adversaries utilize sophisticated snooping equipment to detect timing without actively participating in the network. In this article, a signal processing perspective is presented to study the protection against each kind of adversary, and a path to a unified solution to prevent timing analysis is discussed. Such unified scheduling strategies are shown to require a limited transmission of "dummy" packets to obfuscate the information retrieved by any adversary.



© SHUTTERSTOCK.COM/SERGEY150770

Any communication link in a data network is vulnerable to adversarial timing analysis, where malicious network users observe and analyze the timing of packet transmissions to retrieve critical information about network activity, such as source destination pairs, paths of data flow, hierarchical organization structure, and such like [1]. The availability of such information is a violation of user privacy and could potentially equip an adversary to launch sophisticated network attacks such as denial-of-service [2] and wormholes [3]. Since the early days of World War II [4], timing analysis has proved invaluable in determining the behavioral pattern of network users without actively intruding the network or disrupting network operation. With the advent of the Internet, access to timing information has never been easier, and adversaries utilize both passive and active techniques to retrieve and analyze timing information. Consequently, countermeasures to thwart such information retrieval have also been developed and are prevalent in popular anonymous networking protocols such as Tor (for more information, see www.torproject.org).

Anonymous networking protocols typically rely on randomized packet scheduling policies to counter the effects of timing analysis; these scheduling algorithms obfuscate the timing and the information retrievable there from, but do so at the cost of

Digital Object Identifier 10.1109/MSP.2013.2261131
Date of publication: 20 August 2013

1053-5888/13/$31.00©2013IEEE

network performance as measured by delay and throughput. Consequently, fundamental tradeoffs exist between the degree of privacy achievable and the quality-of-service (QoS) loss [5]. The impact on QoS makes it critical to recognize that different adversarial behavior require different scheduling policies and even more critical to investigate if there exist scheduling policies that provide sufficient privacy from timing retrieval regardless of the nature of an adversary. In this article, a signal processing approach is presented to study the prevention of timing analysis with the aim of unifying the countermeasures against different kinds of adversaries.

In broad terms, adversaries in any networked system can be classified as being either active or passive; active adversaries extract information by masquerading as legitimate users and participating in the network, whereas passive adversaries have access to sophisticated snooping equipment to gather information by monitoring network transmissions silently. In the context of user anonymity from timing analysis, active adversaries transmit packets to a shared router (pretending to be a legitimate user), and extract timing information of a legitimate user using the delays experienced at the shared router. Knowledge of this timing information can then be correlated with known statistical data to determine the specific activity of the legitimate network user. Passive adversaries, in contrast, directly observe transmission timing of packets, using their ability to wiretap communication channels or by using radio detectors in wireless network environments. The timing information thus extracted (actively or passively) is then likely used to gather information about the path of flow of packets from specific users or to specific destinations and, more generally, identify operational patterns in the network. In the ideal system, it should be impossible for an adversary to extract timing information, but in the event that it does happen, the next best scenario is for the network to ensure that timing yields no information about network activity. A secure system ought to be both robust and resilient to attacks.

It is not practical for a router to determine malicious intentions of a network user, and the only means to thwart an active adversary is to modify the scheduling policy of shared routers to minimize the correlation between the delays experienced (by one user) and the timing pattern (of other users). Such policies inevitably result in the outflow of packets from one user to be independent of the arrival pattern of another user. In contrast, scheduling policies to thwart passive adversaries require network routers to match the flow of packets across multiple users sharing the router, thus preventing an adversary from using the observed timing to track any individual flow. Such policies will, expectedly, reveal the timing pattern of flow to an active adversary through the queuing delays. This leads to a vital question: are there policies that minimize the ability of an adversary to extract timing through queuing delays, while at the same time, prevent network information retrieval by eavesdroppers through timing analysis? In broader terms, can a system guarantee sufficient privacy from passive and active timing retrieval? In this article, a signal processing framework is described for this investigation. The approach utilizes statistical inference measures to quantify the effectiveness of scheduling policies against different adversaries and provides the mathematical framework toward developing a unified class of scheduling policies against all timing attacks. The key analytical results in this article focus on the means to counter each kind of adversary individually. The path to unification, while not rigorously delineated, is presented as a class of policies with argumentative support as a means to engage the signal processing community in the pursuit.

A trivial solution to tackle the problem of timing analysis completely would be to fix the transmission schedules of all network routers regardless of arrival timing patterns. Such policies would, however, result in indefinite delays unless packets are dropped [5] or result in an severe loss in link utility in a network [6]. This article will describe a class of policies that require a limited transmission of dummy packets by routers, and as prior work in the area suggests, study of dummy packet insertion is greatly benefited by the statistical inference approach to the problem. To the best of the authors' knowledge, this is the first analytical framework that studies, in unison, privacy from timing analysis against passive and active adversaries.

RELATED WORK

TIMING ANALYSIS

Timing analysis to detect traffic information in computer networks is well documented in literature with some notable examples in [7]–[9]. In [7], the authors showed that a private key can be deciphered by way of a statistical correlation involving key-bit values and the time it takes to perform an encryption algorithm. Similarly, the demonstration of the correlation between timing and the encryption scheme in [8] created the necessity for Netscape to update their browser. In [9], the authors demonstrated that the timing between keystrokes can be used to detect the length and possible character sequences in passwords communicated through the Secure Shell (SSH) protocol.

In any datagram network, knowledge of timing can be used to trace flows of packets and thus can compromise users' anonymity [10]. Specifically, the correlation between incoming and outgoing streams at shared routers induced by the router scheduling policy can be used to track flows from sources to corresponding destinations. On the Internet, senders' anonymity from passive timing analysis is achieved using networks of Chaum mixes [11]. A Chaum mix is a router that waits until packets arrive from multiple users and, thereafter, transmits all the packets in a single batch in a random order, thus reducing the correlation between incoming and outgoing flows. However, the performance of the mix severely deteriorates in presence of resource constraints such as memory and bandwidth and QoS limitations such as delay and throughput. In known implementations of mix-networks, developed primarily to provide anonymity to users, the authors in [12] demonstrated vulnerability to interpacket interval correlation attacks using a realistic traffic model based on HTTP traces. Consequently, several countermeasures have been proposed for detecting and hiding source identities based on flow correlation [13].

The use of delays at a shared router to extract timing information was clearly demonstrated in our recent work [14]. Specifically, we demonstrated in [14] that a side channel does exist in DSL routers, wherein a common scheduler is used to process incoming data packets from all the streams. The attacker sent equally spaced ping packets to the router, addressed to himself, and observed the round trip times (RTTs) of the packets and used that to determine with good accuracy the timing pattern of the legitimate user. The idea of actively extracting information by utilizing a shared resource (specifically a router) and monitoring delays has also been applied to de-anonymize users in [15] where the authors introduce a "queue clogging" attack on anonymous systems. The attack considers a flow that is forwarded by a set of routers in an anonymous communication system. The attacker is able to observe one end of the flow and infers which routers are involved in forwarding it. He does this by sending a large flow to a particular router that "clogs" the router's queue and looks for a corresponding drop in throughput in the anonymous flow. Murdoch and Danezis implemented a version of this attack against popular anonymous networking systems, Tor and MorphMix [16]. Their approach was to send an on-off pattern of high-volume traffic through the anonymous tunnel and a low-volume probe to a router under test. If the waiting times of the probe show a corresponding increase during the "on" periods, the router is assumed to be routing the flow.

THEORETICAL PERSPECTIVES

While covert communication through timing been studied extensively, most notably in [17] and [18], there has been very little work in preventing timing retrieval through delays at shared routers. Most of the solutions proposed to mitigate the timing information leakage are system specific. Some examples are as follows:

- the most common mitigation technique against such channels, cryptographic blinding [19]
- the network pump developed by the Naval Research Lab (NRL) proposed mitigating timing channels that arise in multilevel security systems (MLS), wherein high confidentiality processes can communicate through acknowledgement packets (ACKs) that it sends to low-confidentiality processes [20].

In previous theoretical models to study the impact and prevention of passive timing analysis, the metrics for quantifying anonymity are based on the detection rate (probability of correctly identifying source-destination pairs) or the Shannon entropy of probability distribution of source identities from eavesdropper perspective. These metrics, as well discussed in [5] and [21], consider two distinct models. The first is a short packet burst model, where each packet is treated as an individual entity, and the detection probability or the entropy was used to bound the adversary performance of identifying the source of each individual packet. The second model is the indefinite stream model, where sources transmit infinite length streams of packets to destinations, and the goal was to insert sufficient dummy packets to achieve perfect anonymity at a router for an indefinite period of time. While there are specific applications that benefit from each of these approaches, there is still a need for a unified approach to study anonymity for the general class of finite length packet streams. Furthermore, the theoretical models studied thus far do not directly model an adversary and consequently are not equipped to handle active adversaries capable of capturing packets or modifying input streams to reduce anonymity.

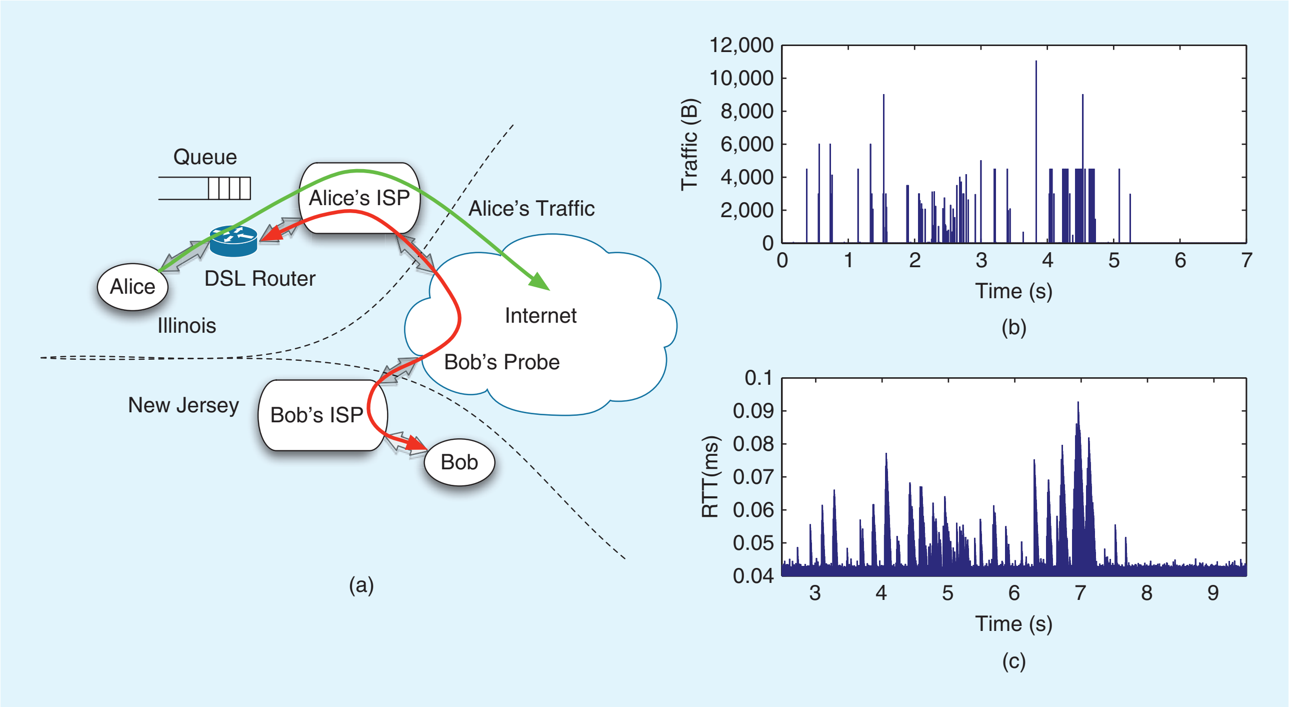
In addition to timing, observable information such as packet lengths can also be used to determining networking information. An information-theoretic approach to preventing such information leakage was discussed in [22].

ACTIVE TIMING EXTRACTION: AN ESTIMATION THEORETIC PERSPECTIVE

Consider the scenario shown in Figure 1. Alice is using her computer at home to connect to the Internet. She connects by using a home digital subscriber line (DSL) router, which connects to a router at her Internet service provider (ISP), connecting to the Internet. The ISP sees all the traffic that Alice sends; however, Alice is not worried because she knows that she is protected by antiwiretapping legislation, and she encrypts all her most sensitive information. Along comes Bob, who is located at another ISP entirely, perhaps even in another country. Bob sends a probe stream to Alice's router. The probes are frequent but small in size. Most importantly, the probes (and responses) make use of a shared queue at Alice's DSL router. As a result, the waiting times of the probes are correlated with Alice's traffic patterns. The traffic entering Alice's computer and Bob's RTTs are shown in Figure 1(b) and (c), and a clear correlation between the two is visible. The fact that DSL routers employed first-come, first-served (FCFS) scheduling aided the attack. In [14], we showed that the high correlation between the two also holds if the DSL router employed a round-robin scheduling scheme instead. Such an attack gives the attacker a noisy observation of the timing and the sizes of packets entering Alice's computer. Although the contents of the packets are not revealed, learning such timing information opens up the possibility for the attacker to carry out remote traffic analysis. In [23], the authors exploit this very side channel to infer the Web site that the victim is visiting. The accuracy of the attack was as high as 70%! Such side channels exist not only in network routers, but in any system with shared resources. In [24], the authors map the internal infrastructure of Amazon's Elastic Cloud Compute (EC2) service and demonstrate that it is possible for an attacker to place his virtual machine (VM) on the same physical computer as the target's VM. They show that once placed on the same physical computer, any timing channel created by sharing of the processor can be exploited by the attacker.

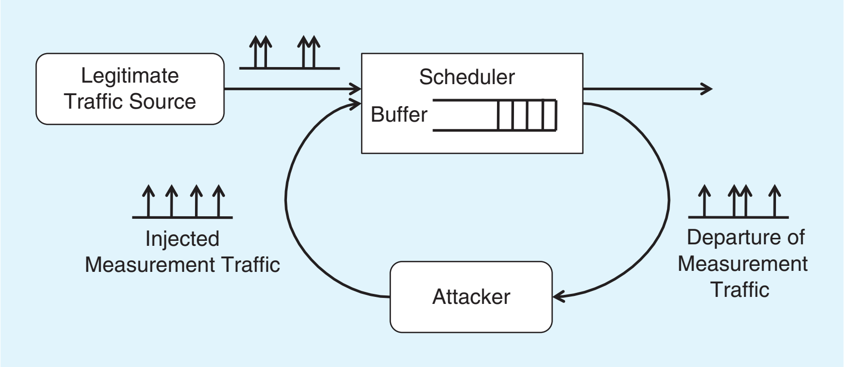
SYSTEM MODEL AND DEFINITIONS

The scheduler is modeled as an infinite buffer server that is serving jobs from two users as shown in Figure 2. The scheduler can serve jobs at a rate of one per unit time. We consider the scenario when one of them is an innocuous user and other a malicious one. The malicious user, Bob, wishes to exploit the queuing side



[FIG1] **A possible exploitation of the timing side channel present inside of a DSL router: (a) side channel setup, (b) DSL traffic entering Alice's router, and (c) round-trip times of Bob's ping packets.**

channel described earlier to learn about the pattern of jobs sent by the innocuous user, Alice. Bob is assumed to know accurately the time when his jobs are issued, and the time it took for the scheduler to process it, i.e., the difference between the completion time of the job and the time when it was issued. Knowing the delays experienced by his jobs, Bob uses this information to guess the arrival pattern of jobs from Alice.



[FIG2] **An event/packet scheduler being exploited by a malicious user to infer the arrival pattern from the other.**

The ability of Bob to successfully learn about Alice's arrival process depends heavily on Alice's arrival process itself. For example, on–off patterns are easier to detect reliably compared to an arrival process that is less bursty. To ensure that the scheduling policies we design are robust to a variety of arrival patterns, Alice's arrival process will be modeled as a Poisson process of rate λ_2, with all the jobs of unit size. We do this partly because Poisson processes are known to have maximum entropy rate among processes of a given rate, and hence represent a rich class of arrival processes, and also because the closed form expressions which can be derived reveal the nature of tradeoffs between privacy and delay.

MEASURING THE STRENGTH OF THE SCHEDULING POLICY: A PRIVACY METRIC

Alice issues unit-sized jobs to the scheduler according to a Poisson process of rate λ_2. The total number of jobs issued by Alice until time u is given by $\mathcal{A}_A(u)$. The malicious user, Bob, also referred to as the attacker, issues his jobs at times $t_1^n \doteq \{t_1, t_2, \ldots, t_n\}$ and is free to choose their sizes, $s_1^n \doteq \{s_1, s_2, \ldots, s_n\}$, as well. We also worked on a version of this problem where the attacker is also forced to issue jobs only of unit size. Many of the results derived here carry over in that scenario as well; refer to [25] and [26]. However, in this scenario, the attacker uses a large amount of bandwidth in just sampling the state of the system, which limits the set of attack strategies available to him. Let $t'^n_1 \doteq \{t'_1, t'_2, \ldots, t'_n\}$, be the departure times of these jobs. Bob makes use of the observations available to him, the set $\{t_1^n, s_1^n, t'^n_1\}$ and the knowledge of the scheduling policy used, in estimating Alice's arrival pattern. The arrival pattern of Alice is the sequence $\{X_k\}_{k=1,2,,N}$, where $X_k = \mathcal{A}_A(kc) - \mathcal{A}_A((k-1)c)$, is the number of jobs issued by Alice in the interval $((k-1)c, kc]$, referred to as the kth clock period of duration c. Nc is the time horizon over which the attacker is interested in learning Alice's arrival pattern.

The privacy offered by a scheduling policy is measured by the long-run estimation error incurred by Bob in such a scenario when he is free to decide the number of jobs he issues, times when he issues them and their sizes, subject to a maximum rate constraint, and when he optimally estimates Alice's arrival pattern. Formally, the timing privacy offered by a scheduling policy is defined to be

$$\mathcal{E}^{c,\lambda_2}_{\text{Scheduling policy}} = \lim_{N\to\infty} \min_{n,t_1^n,s_1^n:\frac{\sum_{i=1}^{n} s_i}{Nc}<1-\lambda_2} \frac{1}{N}\sum_{k=1}^{N} \mathbf{E}[(X_k - \mathbf{E}[X_k \,|\, t_1^n, t'^n_1, s_1^n])^2], \tag{1}$$

where the expectation is taken over the joint distribution of the arrival times of Alice's jobs, the arrival times and sizes of jobs from the attacker, and his departure times. This joint distribution is dependent on the scheduling policy used, which is known to the attacker. Finally, the attacker is assumed to know the statistical description of Alice's arrival process, and he is allowed to pick $\sum_{i=1}^{n} s_i/Nc$, the average rate at which he issues his jobs, to be any value that is less than $1-\lambda_2$, so as to keep the system stable. A scheduling policy is said to preserve user privacy if the resulting estimation error is high.

Our motivation for using the minimum mean squared error (MMSE) as a metric of performance is as follows. The MMSE, as considered in this article, does not conform to a specific adversarial learning technique but serves as a universal lower bound over all adversarial strategies taking into account the complete available information for the entire duration of the system operation. A natural alternative metric would be to measure the information leakage using Shannon's equivocation $\lim_{N\to\infty}(1/N)\sum_{k=1}^{N} H(X_k \,|\, t_1^n, t'^n_1, s_1^n)$. While entropy serves as a measure of uncertainty, which guarantees a minimum probability of error for an adversary (Fano's inequality), MMSE bounds the actual error incurred. The purpose of quantifying privacy is to have a meaningful measure of how breachable a system is, and in that respect, both these measures provide that interpretation. Furthermore, the two metrics are related in the sense that both MMSE and entropy are functionals of the probability mass function of the same conditional random variable, $X_k \,|\, t_1^n, t'^n_1, s_1^n$. MMSE is the variance of the random variable, while equivocation is the entropy of the random variable. Large estimation error does correspond to large entropy, although the relationship between the two is not monotonic. We do note that it is hard for the adversary to know perfectly the joint distribution. Some prior information may be obtained using traffic statistics and knowledge of router resources. The assumption of perfect knowledge of the distribution is a worst case assumption that facilitates the estimation theoretic analysis. The results thus obtained serve as a conservative estimate of the actual MMSE, and the adversary's performance in practice would likely be worse.

In this work, we only consider the case when there are two users of the system: the innocuous user and the attacker. From a privacy perspective, the two-user scenario is the worst case. It is true that if there are more users of the system, the attacker can only learn about the cumulative arrival pattern from all the users. However, as the authors in [24] state, in such systems, the attacker typically waits for a time when he can be assured that the victim is the only other user of the scheduling system and launches an attack then. A policy that fares well on the privacy metric in the two-user scenario is also guaranteed to perform well in the multiple user scenario.

KEY RESULTS

Using the estimation error as the metric for evaluation, some of the key findings are given next with regard to design of scheduling policies.

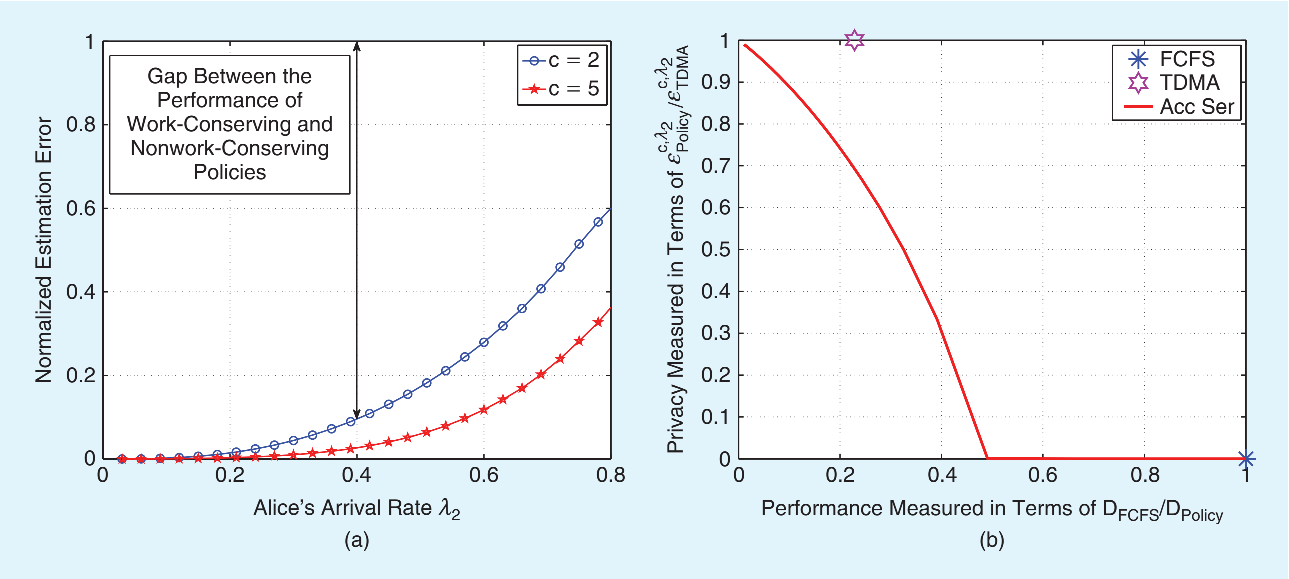
ON THE PRIVACY OF COMMONLY DEPLOYED SCHEDULING POLICIES

FCFS is one of the most commonly deployed scheduling policies owing to its simplicity. It is throughput optimal and results in minimal queuing delay. However, by the nature of the policy, there is a large correlation between the waiting times of jobs of one user and the arrival pattern of the other. Consequently, as shown in [25], by the explicit construction of one attack strategy, FCFS is the weakest policy on the privacy metric. In fact, the attack against the DSL routers was successful because they use the FCFS scheduling policy. On our metric, they offer the least privacy, $\mathcal{E}^{c,\lambda_2}_{\text{FCFS}} = 0$. There exists an attack strategy that guarantees accurate estimation of Alice's arrival pattern.

On the other hand, time-division multiaccess (TDMA), wherein the delays experienced by jobs of one user are completely independent of the arrivals of the other, ranks highest on the privacy metric. As shown in [25], $\mathcal{E}^{c,\lambda_2}_{\text{TDMA}} = \lambda_2 c$ is the highest privacy any scheduling policy can offer. However, TDMA is a highly inefficient policy in terms of throughput and delay, especially when the traffic is varying. It is especially inefficient when the number of users using the scheduler is large.

MAXIMIZING THE PRIVACY OF WORK-CONSERVING POLICIES

Work-conserving (also known as nonidling) policies are those that ensure that the processor never idles as long as there is an unserved job in the system. When all the jobs are of the same size, this is the class of delay-optimal scheduling policies. While FCFS is a work-conserving policy, TDMA is not. In [27], we evaluate the privacy performance of the round-robin scheduling policy, a work-conserving policy that is also known to be fair. Although better than FCFS, we demonstrate that there is a considerable gap between the privacy performance of the round-robin policy (denoted by $\mathcal{E}^{c,\lambda_2}_{\text{RR}}$) and TDMA, thus suggesting that it can be greatly improved upon. The ratio $\mathcal{E}^{c,\lambda_2}_{\text{RR}}/\mathcal{E}^{c,\lambda 2}_{\text{TDMA}}$ is plotted in Figure 3(a). In [27], it is proved that round-robin is a privacy-optimal policy within the class of work conserving policies. Hence, the privacy metric of this policy bounds the privacy metric of every other work-conserving policy. A surprising corollary to this result is that a private source of randomness at the scheduler does not help it, if it is forced to pick a work-conserving policy. For example, consider a policy that randomly switches between



[FIG3] (a) The privacy performance of work-conserving policies and (b) the performance of the idling policy accumulate-and-serve. In (a), the plot of $\mathcal{E}_{\text{RR}}^{c,\lambda_2}/\mathcal{E}_{\text{TDMA}}^{c,\lambda_2}$ is given for the two cases when the clock period is $c = 2$ and $c = 5$. A curve for $\mathcal{E}_{\text{P}}^{c,\lambda_2}/\mathcal{E}_{\text{TDMA}}^{c,\lambda_2}$ lies below $\mathcal{E}_{\text{RR}}^{c,\lambda_2}/\mathcal{E}_{\text{TDMA}}^{c,\lambda_2}$ for any work-conserving policy P.

serving jobs in the FCFS manner to serving jobs in round-robin manner to serving jobs from the user with the longest queue. Because the times when the policy switches behavior is unknown to the attacker, one might expect this policy to outperform deterministic scheduling policies. However, this is not the case. This proves the existence of a fundamental privacy-delay tradeoff in the design of a scheduling policy. If one were to design provably secure scheduling policies, he or she should allow for idling.

DESIGN GENERAL POLICIES THAT OFFER GOOD PRIVACY-DELAY TRADEOFFS

In [25], we design a parametric policy, accumulate-and-serve, which can be tuned to trade off the delay for improved privacy. According to this policy, the scheduler accumulates all the incoming jobs in its buffer for a period of T time units. It then serves all the accumulated jobs from User 1, followed by all the accumulated jobs from User 2, followed by those from User 3, and so on. In the two user scenario discussed before, User 1 could be the attacker, or Alice. If serving all these jobs takes M units of time and if $M < T$, then the scheduler idles for the remaining $T - M$ time before beginning to serve the accumulated jobs. Unlike TDMA, the policy is throughput-optimal. Depending on the value of T, the degree of privacy achieved can be traded off for the average waiting time of the jobs.

In Figure 3(a), we plot $\mathcal{E}_{\text{RR}}^{c,\lambda_2}/\mathcal{E}_{\text{TDMA}}^{c,\lambda_2}$ as a function of λ_2, the arrival rate of jobs from Alice. Note that this is also an upper bound to the privacy performance of any work conserving policy. In the plot, we consider two scenarios, one where the clock period is set to two, and the other where it is set to five. As expected, the attacker incurs a higher normalized error when he wishes to estimate Alice's arrivals with greater precision. Note that there is a relatively large gap between the privacy performance of work conserving and policies that are allowed to idle. For instance, when Alice's arrival rate is less than 0.4, any work-conserving policy can guarantee a privacy no greater than just 10% of the privacy that can be guaranteed by TDMA. In Figure 3(b), we plot the privacy offered by the accumulate and serve policy as a function of the delay offered by it. By changing a parameter of the policy, it can either offer high privacy, like TDMA, but at the cost of an increased delay, or low delay and low privacy, like FCFS.

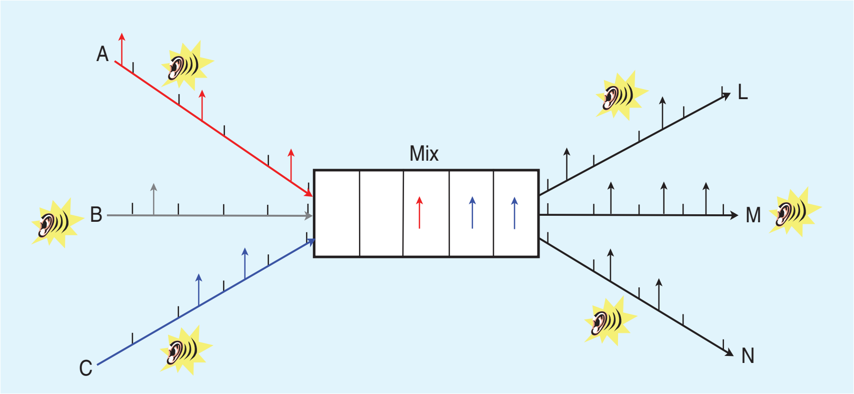
Thus far, through the results in this section, we demonstrated that the estimation model can be used to demonstrate striking properties of known scheduling policies and, further, aid in the design of privacy preserving scheduling policies in the context of active timing extraction. In the next section, we describe the detection model to study passive timing analysis, and subsequently provide the framework for a unified study of thwarting timing attacks.

PASSIVE TIMING ANALYSIS: A DETECTION THEORETIC PERSPECTIVE

Consider the two-user framework as described in the previous section. In the context of passive timing analysis, the task of the router and the capability and knowledge of the adversary are, as expected, quite different from that considered in the section "Active Timing Extraction: An Estimation Theoretic Perspective." The design of the scheduling policy operates under the assumption that timing information is available to an adversary using snooping equipment, and the task is to prevent tracing of flow of packets from sources to destinations.

ROUTER

The router behaves as a Chaum mix, which uses layered encryption and packet padding, so that packets in outgoing streams of the router cannot be linked to packets of incoming streams. The key design task for the router is to randomly reorder the packets



[FIG4] **The router receives packets from users, encrypts and randomly reorders them, and transmits them in their corresponding outgoing link. The adversary observes the arrival and departure processes.**

so that outgoing streams cannot be linked to their corresponding incoming streams (see Figure 4). It is necessary, in this circumstance, to limit the ability of the router to store packets. If unlimited storage is allowed, then it is easily shown that it is optimal for the router to never transmit any packet (a trivial mathematical conclusion but hardly practical).

ADVERSARY

The adversary, no longer an active participant in the system, merely observes the incoming and outgoing streams of the packets and attempts to deduce the source of packets on each outgoing stream. The adversary is aware of limitations of the router, and she also has the ability to capture a limited number of c packets on the incoming links. The effect of capturing a limited number of packets on the achievable anonymity is identical to that accomplished by jamming links for short intervals or by compromising the intermediate nodes between the source and the router. By selectively altering the arrival pattern, the adversary can better detect the out flow of traffic from the router. It is important to note that the adversary is aware of the router's random strategy but does not have access to the realization of the randomness.

PERFORMANCE MEASURE: DETECTION TIME

If an adversary does not observe any arrival or departure process and has no prior knowledge about the sources of outgoing links, the probability of associating the links with any particular source would be the prior probability (in this case $1/u$ for each user). A mixing strategy provides perfect anonymity, if it ensures that probability of predicting the outgoing links of users correctly by the adversary remains $1/u$, independent of the number of packets observed. No mixing strategy can, however, provide perfect anonymity using a limited buffer capacity [28].

In effect, the objective of a memory limited mix is to maintain perfect anonymity for as long as possible, whereas the objective of the adversary is to detect the sources of outgoing packets as quickly as possible, thus representing the scenario for a zero-sum game. It is the length of time that the system can be in perfect anonymity that we designate as the payoff of this zero-sum game. Perfect anonymity ensures that for every observed packet the posterior probability of associating an outgoing stream with any source is equally likely, which would be the case if the adversary had zero observations and no knowledge of the mixing strategy. We aim to characterize this stream length as a function of the system parameters, namely buffer size, rate of dummy transmissions, and capture capacity. This metric will not only serve as a measure of the effectiveness of the mixing strategy but would be of practical utility in determining the allowable length of packet streams from sources.

Let ψ denote a mixing strategy and T_ψ denote the average number of time intervals that Eve requires to violate the perfect anonymity state of the departing packets, where average is taken over arrival process. Let Ψ_m denote the collection of all valid mixing strategies that require a buffer size no greater than m. We define the admissible length $\mathcal{A}_m$ of the router as

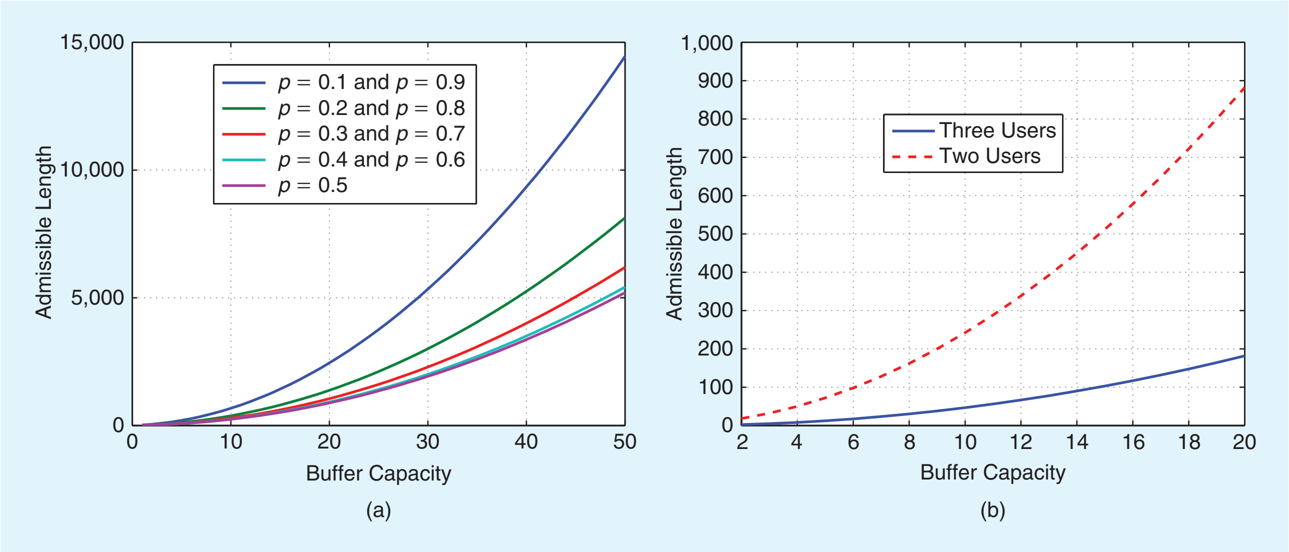
$$\mathcal{A}_m = \sup_{\Psi_m} T_\psi.$$

BASIC RESULTS

In the basic model, the router is not allowed to transmit any dummy packets to append to its traffic pattern, and the adversary cannot capture any packets from the incoming streams of the router.

The router merely stores packets in its buffer and transmits them in a random order to hide their identity while Eve, not having access to the realization of this random ordering, passively observes the arrival and departure processes. The only method the adversary can use to determine the source of an outgoing stream is to look for correlation between incoming and outgoing streams. Consequently, as long as each outgoing stream is equally correlated to both streams, the system is in perfect anonymity. More specifically, as long as the mix ensures that the number of departed packets on any outgoing link is less than the minimum number of arrivals across all incoming links, perfect anonymity is maintained. This idea can be used to prove that the optimal strategy for the mix is to transmit one packet of each user only when packets from all users are present in the buffer, and consequently characterize the admissible length using a recursive solution as in [28, Th. 2]. Further, the theorem also demonstrates that the admissible length varies quadratically in the available memory at the router. Additionally, in contrast to inferences drawn from previous information-theoretic approaches to studying the problem, it is shown that the detection time decreases as the number of sources increases. This surprising result demonstrates the ephemeral nature of the information theoretic measure when dummy transmissions are not allowed.

Figure 5(a) plots the maximum admissible length for two users with equal arrival rate as a function of buffer capacity of the mix for different arrival rates. It is interesting to note from



[FIG5] The admissible length when the mix is not allowed to transmit dummy packets: (a) admissible length for a two-user case as a function of buffer size and (b) admissible length for two users and three users as a function of buffer size.

the theorem that the admissible length increases quadratically with the buffer capacity. Consequently for network applications that do not require long streams of communication (such as Web browsing), the buffer capacity of mixes can be limited sufficiently to prevent traffic analysis. Note that the adversary, as modeled here, is assumed to have complete information about packet arrivals on all links, and a practical adversary with typical noisy observation would have a larger admissible length than that characterized above.

CAPTURE MODEL

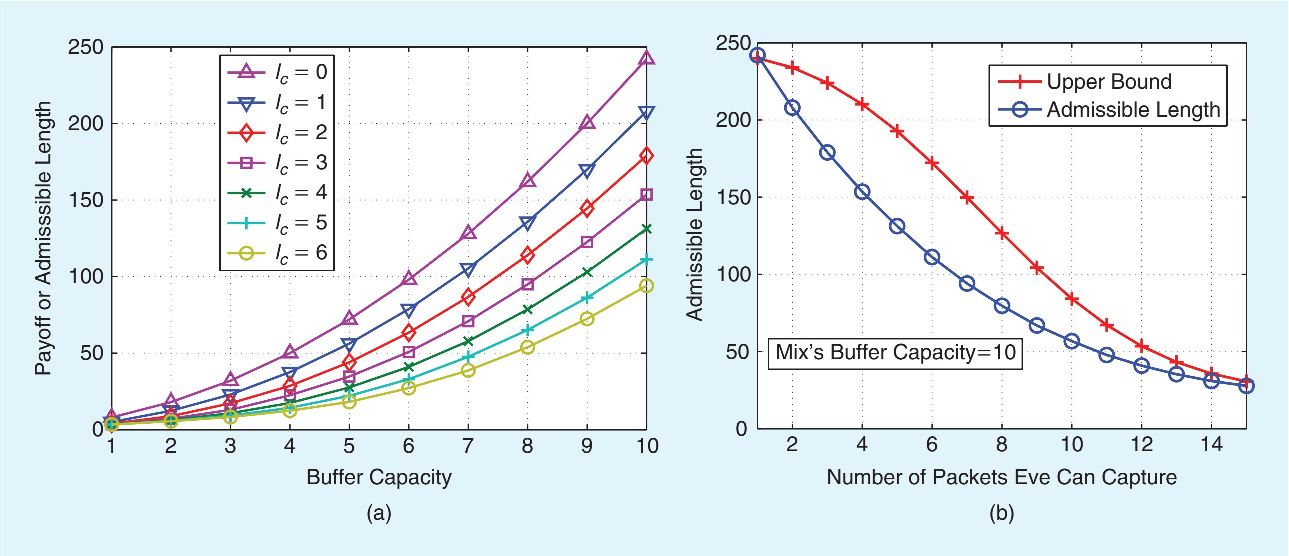
As shown in [28], the signal processing metric that directly measures the adversary's action allows us to study an empowered adversary who captures a limited number of l_c packets of the incoming streams. Such a capture model has previously not been dealt with using information-theoretic models. Note that a limitation on capture ability is important in practice as the adversary needs to ensure that her actions are not detected by intrusion detection applications. As long as the number of packets captured falls within the missed-detection level of the intrusion detector, the adversary can evade detection while compromising anonymity. Since the intrusion detection is designed as part of the network operation, it is fair to assume that scheduling strategies can be designed with knowledge of the capture capacity of the adversary.

It is important to note, however, that as long as the adversary is within the capture limit l_c, the router in unaware if and how many packets are captured. As a result, the optimal strategy of the router would be no different as in the case when the adversary does not capture packets. In other words, the mix has to maintain the condition that no outgoing link has more departures than the number of arrivals in any incoming stream to hide the information of source destination pairs. Consequently, in this two-player game, the optimal strategy for the mix is a dominant strategy. Lets ψ_M denotes this dominant strategy. In [28], we prove that against this dominant strategy, the optimal strategy for the adversary is a threshold strategy, where the adversary decides to capture an arriving packet if and only if the number of packets in the mix's buffer is greater than a threshold. Again, the benefit of using the signal processing model is the ability to characterize the optimal strategy of the adversary, and consequently solve the game theoretic model.

In Figure 6(a), we perform the numerical search and plot the payoff of the mix as a function of buffer capacity for the optimal strategy, and we compare it with the maximum admissible length of the mix when Eve is not allowed to capture packets. It is clear from the figure that Eve can significantly reduce her detection time in determining the source-destination pairs by effectively capturing incoming packets. In Figure 6(b), we plot the numerically computed admissible length of the model $\mathcal{G}_E$ as a function of maximum number of packets Eve can capture and compare it against the upper bound. As is evident from the figure, the upper bound provides a good approximation of the optimal threshold strategy when the maximum number of packets Eve can capture is greater than the buffer capacity of the mix.

TOWARD A UNIFIED SCHEDULING STRATEGY: INTRODUCING DUMMY TRANSMISSIONS

We observe that the strategies specified in the previous two sections are quite different (expectedly as they deal with different kinds of adversaries). Specifically, if the "mix" router were to constantly match the flow of packets across both users, then the active adversary would be able to identify the arrival time of each packet from the legitimate user within any desirable clock interval by transmitting packets matching its own clock periods of interest. In contrast, if the router were to adhere to an accumulate and serve policy to minimize active timing extraction, then the unmatched output flows would reveal, within a single accumulate period, the source of outgoing packets with high probability, thus limiting the ability of the router to serve larger flows in the presence a passive adversary. It is arguable that protection



[FIG6] The admissible length when the adversary can capture packets: (a) admissible length as a function of the capture limit l_c and (b) upper bound and optimal payoff.

from an adversary with sophisticated snooping equipment should suffice, and additional protection from timing extraction through packet insertion would be redundant. Note that while "mix" routers prevent the tracing of flows from sources to destinations, active adversaries can still determine the timing pattern of the source transmissions, which alone, as demonstrated in [7]–[9], can provide invaluable information to malicious adversaries.

A key benefit of the signal processing-based approach over previous entropy-based approaches is the analysis when the router is allowed to transmit limited number of dummy packets to mask its traffic pattern. Herein, an inverse linear relationship between the detection time for passive adversaries and the rate of dummy transmissions has been shown in [28]. In particular, for a simple two-user case the required rate of dummy transmissions to maintain perfect undetectability was shown to be $d_r = 1/(2m + 1)$, where m is the memory size of the router. It was further shown that the analysis can be extended to the capture model discussed previously for an empowered adversary by providing lower and upper bounds on the relationship between dummy rate and the detection time. The main inference on the equilibrium strategies for the scheduler and the adversary do not change with the addition of dummy packets. The inclusion of dummy transmissions, at the cost of reduced throughput, allows the modification of the optimal scheduling policy against passive adversaries to closely mirror the accumulate-and-serve policy required against active adversaries thus bridging the gap between the two problems.

Specifically, the router waits until it has accumulated packets from both users together not exceeding the buffer capacity. At this point, the router evaluates the allowable dummy transmissions (a predetermined limit on the dummy transmission rate is imposed to satisfy bandwidth utilization requirements). If the allowable number of dummy packets can be used to supplement the difference in arrivals across the users, then all packets are transmitted in a manner that neither reveals the pattern of arrivals of one user nor compromises the anonymity from the perspective of a passive adversary. If the allowable dummy transmissions do not suffice then some packets are held back from the user with higher number of arrivals so as to maintain the anonymity. The router then waits for the buffer to fill up and repeats this process.

It is easy to see that under this modified scheduling strategy, the admissible length achieved in Theorems 1 and 4 in [28] does not change thus maintaining the anonymity from passive adversary. As the rate of dummy transmissions increase, the admissible length increases (in an inverse linear manner) until it reaches the limit of perfect anonymity. When the strategy is utilized against an active adversary sharing a router, the dummy transmissions can only increase the estimation error as specified in [29, Th. 4.2]. This is easily seen from the assumptions of Theorem 4.2, wherein the adversary is assumed to estimate the total number of packets in each accumulate period accurately. The addition of dummy packets would only increase the uncertainty in this estimate, thus increasing the privacy from an active adversary. Consequently, as the rate of dummy transmissions increase, the privacy from both kinds of adversaries will increase toward their respective limits.

The signal processing approach not only provides a unified analytical framework to study privacy from timing analysis but also treats the information hiding problem from the adversary perspective; the detection and estimation metrics measure the performance of the best possible adversary directly (as opposed to entropic measures, which indirectly bound the performance). Consequently, the approach allows for further analysis when adversarial capabilities are expanded to include capture of packets transmitted by a legitimate user or alter the timing pattern and reduce detection time or estimation error.

CONCLUSIONS

To summarize, in this article, we describe the protective mechanisms against timing analysis from two broad classes of adversaries: active adversaries who use a shared router to determine transmission schedules of a legitimate user and a passive eavesdropper who uses snooping equipment to gather timing information. We believe that the inference theoretic models provide the right framework to study this problem and design policies that offer sufficient protection while not sacrificing significantly on the performance. We further believe that, despite stark differences in policy design between different classes of adversaries, the use of dummy transmissions can help bridge the gap toward a unified design of scheduling policies to thwart timing analysis.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation through grants CNS-1117701, CCF-1149495, CCF 10-54937 CAR, CCF 10-65022 and the Air Force Office of Scientific Research under grants FA9550-11-1-0016 and FA9550-10-1-0573.

AUTHORS

Sachin Kadloor (kadloor1@illinois.edu) graduated from the Indian Institute of Technology, Madras, with a bachelor's degree in electrical engineering in 2007. He received a master's degree in 2009 from the University of Toronto. His research dealt with power allocation in selection-based cooperative cellular networks. Since September 2009, he has been working toward his Ph.D. degree at the University of Illinois, Urbana-Champaign. His current research interests are queuing theory and information theory pertaining to timing channels and issues in network security. He is a Student Member of the IEEE.

Parv Venkitasubramaniam (parv.v@lehigh.edu) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, in 1998 and the M.S and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, New York, in 2005 and 2008, respectively. He is an assistant professor in the Electrical and Computer Engineering Department at Lehigh University. His research interests include security and anonymity in networks, information theory, distributed signal processing, and smart energy distribution. He received the 2004 Leonard G. Abraham Award from the IEEE Communication Society and a National Science Foundation CAREER Award in 2012. He is a Member of the IEEE.

Negar Kiyavash (kiyavash@illinois.edu) received the B.S. degree in electrical and computer engineering from the Sharif University of Technology, Tehran, in 1999, and the M.S. and Ph.D. degrees, also in electrical and computer engineering, from the University of Illinois at Urbana-Champaign in 2003 and 2006, respectively. From 2006 through 2008, she was a member of the research faculty in the Department of Computer Science and a research scientist at the Information Trust Institute, both at the University of Illinois at Urbana-Champaign. Her research interests are in information theory and statistical signal processing with applications to computer, communication, and multimedia security. She is a Senior Member of the IEEE.

REFERENCES

[1] J.-F. Raymond, "Traffic analysis: Protocols, attacks, design issues and open problems," in *Designing Privacy Enhancing Technologies: Proc. Int. Workshop Design Issues Anonymity and Unobservability*, 2001, pp. 10–29.

[2] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in *Proc. 2003 Conf. Applications, Technologies, Architectures, Protocols for Computer Communications*, 2003, pp. 99–110.

[3] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Wormhole attacks in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 2, pp. 370–380, 2006.

[4] N. West, *The SIGINT Secrets: The Signal Intelligence War: 1900 to Today*. New York: William Morrow, 1988.

[5] P. Venkitasubramaniam, T. He, L. Tong, S. Wicker, and C. University, "Toward an analytical approach to anonymous wireless networking," *IEEE Commun. Mag.*, vol. 46, pp. 140–146, Feb. 2008.

[6] B. Radosavljevic and B. Hajek, "Hiding traffic flow in communication networks," in *Proc. Military Communications Conf.*, 1992, pp. 1096–1100.

[7] E. English and S. Hamilton, "Network security under siege: The timing attack," *Computer*, vol. 29, pp. 95–97, Mar. 1996.

[8] I. Goldberg and D. Wagner, "Randomness and the Netscape browser," *Dr. Dobb's*, vol. 21, no. 1, pp. 66–71, 1996.

[9] D. X. Song, D. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on SSH," in *Proc. 10th USENIX Security Symp.*, 2001, p. 25.

[10] A. Back, U. Möller, and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity providing systems," in *Proc. 4th Int. Workshop Information Hiding, (IHW '01)*, London, U.K., pp. 245–257.

[11] D. Chaum, "Untraceable electronic mail, return addresses and digital pseudonyms," *Commun. ACM*, vol. 24, pp. 84–88, Feb. 1981.

[12] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: Attacks and defenses," in *Computer Security ESORICS 2006* (Lecture Notes in Computer Science, vol. 4189), D. Gollmann, J. Meier, and A. Sabelfeld, Eds. Berlin: Springer, 2006, pp. 18–33.

[13] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, "Timing attacks in low-latency mix systems (extended abstract)," in *Proc. 8th Int. Financial Cryptography Conf. (FC 2004)*, Key West, FL, LNCS 3110, pp. 251–265.

[14] S. Kadloor, X. Gong, N. Kiyavash, T. Tezcan, and N. Borisov, "Low-cost side channel remote traffic analysis attack in packet networks," in *Proc. IEEE ICC 2010—Communication and Information System Security Symposium*, Cape Town, South Africa, pp. 1–5.

[15] A. Back, U. Möller, and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity providing systems," in *Information Hiding* (Lecture Notes in Computer Science, vol. 2137). New York: Springer-Verlag, 2001, pp. 245–247.

[16] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," in *Proc. 2005 IEEE Symp. Security and Privacy (SP '05)*, Washington, D.C., pp. 183–195.

[17] V. Anantharam and S. Verdu, "Bits through queues," *IEEE Trans. Inform. Theory*, vol. 42, pp. 4–18, Jan. 1996.

[18] J. Giles and B. Hajek, "An information-theoretic and game-theoretic study of timing channels," *IEEE Trans. Inform. Theory*, vol. 48, pp. 2455–2477, Sept. 2002.

[19] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology: Proc. Crypto*, New York: Plenum, 1983, vol. 82, pp. 199–203.

[20] I. S. Moskowitz and A. R. Miller, "The channel capacity of a certain noisy timing channel," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1339–1344, July 1992.

[21] C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proc. Privacy Enhancing Technologies Workshop (PET 2002)*, LNCS 2482, pp. 54–68.

[22] S. Mathur and W. Trappe, "BIT-TRAPS: Building information-theoretic traffic privacy into packet streams," *IEEE Trans. Inform. Forensics Sec.*, vol. 6, no. 3, pp. 752–762, 2011.

[23] X. Gong, N. Borisov, N. Kiyavash, and N. Schear, "Website detection using remote traffic analysis," in *Privacy Enhancing Technologies* (Lecture Notes in Computer Science, vol. 7384). Berlin: Springer, 2012, pp. 58–78.

[24] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds," in *Proc. 16th ACM Conf. Computer Communications Security (CCS '09)*, New York, pp. 199–212.

[25] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam, "Mitigating timing based information leakage in shared schedulers," in *Proc. IEEE 2012 INFOCOM*, pp. 1044–1052.

[26] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam. Scheduling with privacy constraints. presented at the 2012 IEEE Information Theory Workshop (IEEE ITW 2012) [Online]. Available: http://www.ifp.illinois.edu/~kadloor1/kadloor_itw_extended.pdf

[27] S. Kadloor and N. Kiyavash, "Delay optimal policies offer very little privacy," in *Proc. 32nd Annu. IEEE Int. Conf. Computer Communications (INFOCOM'2013)*, Turin, Italy, pp. 2552–2560.

[28] A. Mishra and P. Venkitasubramaniam, "Admissible length study in anonymous networking: A detection theoretic perspective," *IEEE J. Select. Areas Commun.*, to be published.

[29] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam. Mitigating timing based information leakage in shared schedulers. *IEEE Trans. Networking* [Online]. submitted for publication. Available: http://www.ifp.illinois.edu/~kadloor1/kadloor2013ton.pdf