On the Effectiveness of Continuous-Time Mixes under Flow-Correlation Based Anonymity Attacks

Ye Zhu[1], Xinwen Fu[2], and Riccardo Bettati[3]
(Corresponding author: Ye Zhu)

Department of Electrical and Computer Engineering, Cleveland State University[1]
2121 Euclid Avenue, SH 433, Cleveland, OH 44115-2214, USA (Email: y.zhu61@csuohio.edu)
College of Business and Information Systems, Dakota State University[2]
Madison, SD 57042, USA
Department of Computer Science, Texas A&M University[3]
College Station TX 77843-3112, USA

(Received Sep. 17, 2006; revised and accepted Nov. 8, 2006)

Abstract

In flow-based mix networks, flow correlation attacks have been proposed earlier and have been shown empirically to seriously degrade mix-based anonymous communication systems. In this paper, we theoretically analyze the effectiveness of a mix network under flow correlation attacks. Our formulae clearly show how a mix network will ultimately fail when an adversary has access to sufficiently long flow samples, independently of the type of flows (TCP or UDP). We illustrate the analysis methodology by modelling a continuous-time mix, which randomly delays each incoming packet. Our queuing-model-based analysis can provide useful guidelines for designers who develop and deploy anonymity systems.

Keywords: Anonymity, mix, mutual information

1 Introduction

As the popularity of the Internet grows, people have raised more and more concerns over information privacy. Anonymity is feasible and beneficial in many scenarios, such as privacy-preserving Web browsing, electronic voting, and many other e-business applications. The nature of many such applications requires that the identities of participants remain confidential from either other participants or from a third party.

Achieving anonymity in open environments such as the Internet is a challenging problem. Encryption alone cannot preserve the anonymity of communication, since the identities or locations of participants can be easily inferred from the packet headers. Additional measures such as rerouting must be put in place to hide the identities of participants.

Chaum [3] proposed the use of special proxies, called as *mixes*, to relay messages for anonymous email applications. A mix may delay, batch and reorder packets to disrupt the packet-level timing correlation of packets into and out of the mix. Multiple mixes form a *mix network*, in which a sender chooses a path through the mix network to the receiver. In general the sender uses source routing and encrypts messages in an onion-like way [14]. Each intermediate mix gets the address of the next mix after decrypting the message and relays the "thinner" stripped message to the next mix with its own address as the source address. Researchers have extended *message-based* mix networks (primarily used for email applications) to *packet-based* mix networks for low-latency, flow-based applications, such as Web browsing and other delay-sensitive data transfer applications. Clearly, attacks to message-based mix networks are effective against these low-latency mix networks as well. Mix networks have been found to be susceptible to a number of attacks such as the active trickle and flooding attacks [7, 11]), passive packet counting attack [1] and various forms of correlation and intersection attacks [5, 13, 15].

We previously proposed a class of statistical attacks that exploit flow-level timing signatures [15]. These so-called *flow correlation attacks* can seriously degrade flow-based anonymity communication systems in particular if they carry TCP flows. The idea underlying the flow correlation attack is: Assuming an adversary intercepts a sample of Alice's flow going through a mix and samples of the mix's output aggregate flows, the adversary can determine the output link for Alice's flow by comparing the dependence between Alice's flow sample to the aggregate flow samples. We use mutual information to measure

the dependence. The intuition of flow correlation attacks is: The mix network transforms the Alice's flow through different mixing strategies, but the actual outgoing link for Alice's flow will be dependent on Alice's flow while Alice's flow is independent of the other outgoing links. Depending on the attacker's capability, the attack can be performed at different levels: In a global passive attack, (where the attacker can observe all the links in a mix network), the adversary can determine Alice's flow path by determining the mix's output link step-by-step. We will see that there is no need to have access to Alice's flow inside the network for this step-by-step analysis, as Alice's flow characteristics used in this attack are largely invariant across mixes, in particular if it is a TCP flow. If the adversary can only observe a part of mix network, the adversary can aggregate mixes into a *supermix* so that she can observe all the outgoing links of the supermix. Then the adversary can perform the flow correlation attack at the supermix level.We use *detection rate* [15], the probability of identifying the actual outgoing link, as a metrics to evaluate the performance of mixes under flow correlation attacks. Experiments have shown [15] that flow correlation attacks are effective even with large amounts of cross traffic.

In this paper, we formally model the effectiveness of flow correlation attacks. Our formulae clearly show the relationship between detection rate and the amount of available data for a mix network. Our results are applicable to mix networks of any size and topology and mix networks using any batching strategies. They are also applicable to any type of traffic. As expected, mix networks will ultimately fail when an adversary has access to sufficiently long[1] flow samples. Our analytical framework can be used to analyze any mixing strategies [11] under flow correlation attacks. Specially, we show how to use a combination of $M/M/\infty$ and $M/D/1$ queuing models to model the continuous-time mix, in which packets are randomly delayed. We will show that the detection rate for Poisson traffic models is a lower bound for TCP traffic, which is dominant in the Internet. We believe that our analysis based on queuing models can provide a useful guideline for designers to develop and deploy anonymity systems, as (a) it can be used in a similar way for a variety of statistical traffic analysis attacks, and (b) it is applicable to a variety of mixes.

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 describes the mix network model and threat model. Section 4 introduces the flow correlation attack. Section 5 describes the modelling framework of flow correlation attack and its use on the continuous-time mix. Section 6 uses experiments and simulations to validate our theory and also evaluates the impact of the parameters of continuous-time mix on the performance of the continuous-time mix. We conclude the paper and discuss the future work in Section 7.

[1]In fact, an adversary only needs 10 seconds of traffic to achieve a detection rate of 95% in our study. Typical TCP flows such as in downloads, last significantly longer!

2 Related Work

For anonymous email applications, Chaum [3] proposed to use relay servers, called *mixes*, which reroute messages that are encrypted by the public keys of the mixes. An encrypted message is analogous to an onion constructed by a sender, who sends the onion to the first mix. Using its private key, the first mix peels off the first layer. Inside the first layer is the second mix's address and the rest of the onion, which is encrypted with the second mix's public key. After retrieving the second mix's address, the first mix forwards the peeled onion. This process proceeds until the core part of the onion is forwarded to the receiver.

More recently *low-latency, flow-based* anonymous communication systems have been proposed, which forward individual packets of flows instead of entire messages. Low-latency anonymous communication can be divided into systems using *core mix networks* and *peer-to-peer networks*. In a system using a core mix network, users connect to a pool of mixes, which provides anonymous communication, and users select a forwarding path through this core network to the receiver. *Onion routing* [14] and *Freedom* [2] belong to this category. In a system using a peer-to-peer network, every node in the network is a mix, but it can also be a sender and receiver. *Crowds* [10] and *Tarzan* [6] belong to this category.

This paper is interested in the study of passive traffic analysis attacks against low-latency anonymous communication systems. Sun *et al.* Serjantov and Sewell [12] analyzed the possibility of a lone flow along an input link of a mix. If the rate of this lone input flow is roughly equal to the rate of a flow out of the mix, this pair of input flow and outflow flow can be easily correlated. Other analysis focus on the anonymity degradation when some mixes are compromised, e.g. [10].

To find out the path Alice's flow takes through a mix, an adversary may measure the dependency between each output link traffic and Alice's flow. We previously proposed using mutual information for the dependency measurement [15]. In the single-mix case, an adversary collects samples from an input flow and each output flow of the mix. Each sample is divided into multiple equally-sized segments based on time. The number of packets in each segment is counted and forms a time series of packet counts. Then the adversary chooses the output link whose flow's packet count time series has the biggest mutual information with the input flow's packet count time series as the input flow's output link.

Levine *et al.* [9] are also interested in a similar problem, where the mixes are assumed to be compromised so that the adversary can get timing information of each individual flow through the compromised mix. In this attack, cross correlation is used to measure similarities between the flows through compromised mixes to determine the path taken by a flow. This attack differs from the flow correlation attack described above in that it gives the attacker access to per-flow timing information, while the flow correlation attack has access to timing information

for *aggregate* flows only.

The concept of continuous-time mix is introduced by Danezis in [5]. Danezis proved that the optimal mix strategy that maximizes anonymity is the *Exponential Mix*, i.e. a Stop-and-Go Mix that delays packets individually according to exponential distribution. This paper will focus on the Exponential Mix. Danezis also proposed an attack on the continuous mix that is based on likelihood ratio testing. This particular attack assumes accurate synchronization between the trace data of the flow traffic and that of the other traffic in network. Such accurately synchronized data may be difficult to get hold of, as timing data may be gathered from a variety of different sources, such as NetFlow traces from routers, or snooping on links. In comparison, flow correlation attacks perform well on poorly synchronized data.

3 Models

Flow-based Mix Network Model: A flow-based mix is a relay server for the packets of anonymous flows. A mix operates as follows: (1) the sender attaches the receiver address to a packet and encrypts the entire packet with the mix's public key; (2) the mix collects a batch of packets (from different senders), and decrypts each of them separately to obtain the receiver addresses; (3) finally the mix sends the decrypted packets out in a rearranged order to the various receivers. Batching and reordering are necessary techniques for a mix to prevent traffic analysis attacks, which may correlate input packets and output packets by their timing.

A mix network consisting of multiple mix servers can provide enhanced anonymity. In a mix network, senders route their packets through a series of mixes. Therefore, even if an adversary compromises one mix and discovers the correlation between its input and output packet flows, other mixes along the path can still provide an adequate level of anonymity. Figure 1 gives an example of a mix network. Alice selects a series of mixes (depicted in gray) through the mix network to communicate with Bob.

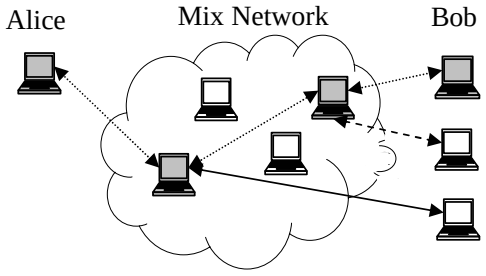


Figure 1: A mix network

In this paper, we investigate the anonymity of flow-based anonymity systems that use the continuous-time mixes. In a continuous-time mix [5, 8], each packet into the mix is assigned a delay (deadline) that satisfies some given distribution. The packet is sent out when its deadline is reached. Danezis [5] proves that in terms of an entropy-based anonymity metric, the packet delay should satisfy an exponential distribution, which is also the setting used in this work.

Threat Model: In the following, we summarize the adversarial assumptions considered in this paper: *(a)* The content of communication between legal participants is protected by underlying encryption algorithms and is not accessible to attacks. *(b)*The adversary is an external one, and therefore is not a legal participant. *(c)* The adversary can passively eavesdrop on the communication session. *(d)* The adversary is global: she can observe the traffic on any link in the system.

These assumptions are widely used in open environments to evaluate anonymity systems aimed at achieving strong anonymity.

4 Flow Correlation Attacks

We formalize the flow correlation attack [15] as follows:

1) **Capture the timing data of the flows:** The adversary captures samples of Alice's flow and the aggregate flows on each possible outgoing link. Each sample (of length t) is divided into segments of equal length, based on time. The length of each segment is denoted as the *sampling interval*, and t is the *sample length*. In our paper, we use a sampling interval of 10ms which captures the transmission dynamics of TCP accurately, especially for TCP flows. The number of packets in each segment is counted and forms a time series of packet counts for the flow sample.

   Each time series has a size of N, denoted as *sample size*. We have a set of M possible outgoing links. We denote Alice's flow packet count time series as X and the aggregate flow packet count time series as $Y_1, \cdots, Y_M$ for the M possible outgoing links.

2) **Measure the similarity:** The adversary can use *mutual information* [4] in Equation (1) to measure the dependency between flows:

$$\hat{I}(X;Y) = \sum_{u=0}^{r}\sum_{v=0}^{s} \hat{p}_{uv} \log \frac{\hat{p}_{uv}}{\hat{p}_{u.}\hat{p}_{.v}}, \qquad (1)$$

   where $\hat{p}_{u.}$ denotes the *frequency* of $X = u$, $\hat{p}_{.v}$ denotes the *frequency* of $Y = v$, $\hat{p}_{uv}$ denotes the *joint frequency* of $(X, Y) = (u, v)$, and r and s are ranges of the number of packets in a segment for Alice's flow and aggregate flows respectively.

   If we assume that Alice's flow sample creates a time series $(u_1, \cdots, u_N)$, and the j^{th} aggregate flow sample creates a time series $(v_{j,1}, \cdots, v_{j,N})$, then one sample of the joint distribution $p(u, v_j)$ of (X, Y_j) is as follows:

$$\text{one sample of } p(u, v_j) = ((u_1, v_{j,1}), \cdots, (u_N, v_{j,N})). \qquad (2)$$

In the following, we use $\hat{p}$ to denote frequency, while p denotes the actual underlying distribution. Similarly in the rest of paper, we use $\hat{I}$ to denote mutual information estimation based on the frequency $\hat{p}$ as in Equation (1) and I to denote the actual mutual information, which is based on the actual underlying distribution p. More specifically we use I_j and $\hat{I}_j$ to denote actual mutual information and mutual information estimation between Alice's flow sample and the j^{th} aggregate flow sample.

3) **Decide on the receiver:** To decide the outgoing link of Alice's flow, the adversary chooses the one whose aggregate flow sample time series has the biggest mutual information with Alice's flow sample time series. That is, if $\hat{I}(X, Y_i) > \hat{I}(X, Y_j), j \neq i, 1 \leq j \leq M$, then Alice's flow goes through the i^{th} link.

In the following section, we build an analytical framework to analyze *(a)* why flow correlation attacks succeed and *(b)* the system parameters' impact on the performance of flow correlation attacks. The framework of anonymity analysis in this paper can be easily extended to other mixing strategies in [11] although we focus on continuous-time mixes with exponentially distributed packet delay.

5 Modelling Flow Correlation Attacks

5.1 Detection Rate as Anonymity Degree

In this paper, we measure the anonymity degree in form of *detection rate*, which is defined as the probability that the adversary correctly recognizes the actual outgoing link.

Without loss of generality, we denote the packet count on the correct outgoing link as Y_1 and packet counts on the other links as $Y_2, \cdots, Y_M$. We also denote the mutual information between X and Y_1 as $\hat{I}_1$, and the mutual information between X and the other packet counts as $\hat{I}_2, \cdots, \hat{I}_M$. The detection rate D can be calculated as follows:

$$D = Pr(\hat{I}_1 > \hat{I}_2, \cdots, \hat{I}_1 > \hat{I}_M). \tag{3}$$

5.2 Distribution of Mutual Information Estimation

Flow correlation attacks rely on establishing statistical properties of traffic based on collected data. The effectiveness of such attacks therefore depends directly on the accuracy of the attackers' estimation techniques. For the flow correlation attack described above, we must assess the accuracy of the attacker's estimation of the mutual information. In this section, we prove a few characteristics about estimation of mutual information. This will be of use in the following sections.

According to Central Limit Theorem, when the sample size N is sufficiently large, mutual information estimation satisfies a normal distribution. So to obtain the distribution function, we only need to estimate normal distribution estimation's mean and variance, which are given in Lemma 1 and 2, respectively.

Lemma 1. *The mean of the mutual information* estimation $\hat{I}_j$ *is given by* $E(\hat{I}_j) \approx I_j + \frac{(r-1)(s-1)}{2N}$ *where* I_j *is the* actual *mutual information between Alice flow packet count and the* j^{th} *possible aggregate flow packet count, and* r *and* s *are the range of the number of packets in a segment for Alice's flow and the* j^{th} *possible aggregate flow respectively.*

Proof. We estimate the mutual information I_j as defined in Equation (1) as follows,

$$\begin{aligned}\hat{I}_j &\approx \sum_{u=0}^{r}\sum_{v_j=0}^{s} \hat{p}(u, v_j) \log \frac{\hat{p}(u, v_j)}{\hat{p}(u)\hat{p}(v_j)} \\ &= \sum_{u, v_j} \hat{p}(u, v_j) \log \hat{p}(u, v_j) \\ &\quad - \sum_{u} \hat{p}(u) \log \hat{p}(u) \\ &\quad - \sum_{v_j} \hat{p}(v_j) \log \hat{p}(v_j). \end{aligned} \tag{4}$$

.

If we apply a second-order Taylor expansion to the three items in Equation (5.2) at $p(u, v_j)$, $p(u)$, and $p(v_j)$, respectively, after a series of rearrangements, we have

$$\begin{aligned}\hat{I}_j &= \sum_{u, v_j} \hat{p}(u, v_j) \log \frac{p(u, v_j)}{p(u)p(v_j)} \\ &\quad + \frac{1}{2} \sum_{u, v_j} \frac{1}{p(u, v_j)} ([\hat{p}(u, v_j) - p(u, v_j)]^2 \\ &\quad - \frac{1}{2} \sum_{u} \frac{1}{p(u)} [\hat{p}(u) - p(u)]^2 \\ &\quad - \frac{1}{2} \sum_{v_j} \frac{1}{p(v_j)} [\hat{p}(v_j) - p(v_j)]^2. \end{aligned} \tag{5}$$

Now we are ready to compute the mean of $\hat{I}_j$ as follows:

$$E[\hat{I}_j] = \sum_{\substack{n_{0,0}, \cdots, n_{r,s} \\ n_{0,0} + \cdots + n_{r,s} = N}} p(n_{0,0}, \cdots, n_{r,s}) \hat{I}_j, \tag{6}$$

where n_{u,v_j} is the frequency of (u, v_j). One sample in Equation (2) corresponds to a $(n_{00}, \cdots, n_{rs})$, which gives one possible mutual information estimation. $p(n_{00}, \cdots, n_{rs})$ satisfies a multinomial distribution.

Substituting Equation (5) into Equation (6) and using multinomial distribution's formulae for mean and vari-

ance, we have, after rearrangements,

$$\begin{aligned} E[\hat{I}_j] &= \sum_{u,v_j} p(u,v_j)\log\frac{p(u,v_j)}{p(u)p(v_j)} \\ &\quad +\frac{1}{2N}\sum_{u,v_j}(1-p(u,v_j)) - \frac{1}{2N}\sum_{u}(1-p(u)) \\ &\quad -\frac{1}{2N}\sum_{v_j}(1-p(v_j)) \\ &= I_j + \frac{(r-1)(s-1)}{2N}. \end{aligned}$$

□

Lemma 2. *The variance of the mutual information estimation $\hat{I}_j$ is given by $var(\hat{I}_j) \approx \frac{C_j}{N}$, where C_j is an expression and is defined as follows:*

$$\begin{aligned} C_j &= \sum_{u,v_j} p(u,v_j)\left(\log\frac{p(u,v_j)}{p(u)p(v_j)}\right)^2 n \\ &\quad -\left(\sum_{u,v_j} p(u,v_j)\log\frac{p(u,v_j)}{p(u)p(v_j)}\right)^2, \end{aligned} \tag{7}$$

where $p(u,v_j)$ is the original probability distribution of (X,Y_j).

Proof. To obtain the variance of $\hat{I}_j$, we perform an approximation by only keeping the first item in Equation (5). Thus,

$$\hat{I}_j \approx \sum_{u,v_j}\hat{p}(u,v_j)\log\frac{p(u,v_j)}{p(u)p(v_j)}.$$

Since $\hat{p}(u,v_j) = \frac{n_{u,v_j}}{N}$, we have

$$\hat{I}_j = \frac{1}{N}\sum_{u,v_j} n_{u,v_j}\log\frac{p(u,v_j)}{p(u)p(v_j)}. \tag{8}$$

The multinomial distribution has the following property:

$$\begin{aligned} &\sum_{u,v_j} s_{u,v_j} n_{u,v_j} \\ &= N\left(\sum_{u,v_j} p(u,v_j)s_{u,v_j}^2 - (\sum_{u,v_j} p(u,v_j)s_{u,v_j})^2\right), \end{aligned}$$

where s_{u,v_j} is a constant. Applying this property to Equation (8) with

$$s_{u,v_j} = \log\frac{p(u,v_j)}{p(u)p(v_j)}.$$

We have

$$\begin{aligned} Var[\hat{I}_j] &\approx \frac{1}{N^2}Var\left[\sum_{u,v_j} n_{u,v_j}\log\frac{p(u,v_j)}{p(u)p(v_j)}\right] \\ &= \frac{1}{N}\sum_{u,v_j} p(u,v_j)\left(\log\frac{p(u,v_j)}{p(u)p(v_j)}\right)^2 \\ &\quad -\frac{1}{N}\left(\sum_{u,v_j} p(u,v_j)\log\frac{p(u,v_j)}{p(u)p(v_j)}\right)^2 \\ &= \frac{C_j}{N}. \end{aligned}$$

□

5.3 Detection Rate Theorem

Based on the characteristics of the mutual information estimation determined above, we can calculate the detection rate using the following Theorem 1. The intuition for the proof of this theorem is: If Alice's flow X goes through the first outgoing link, the aggregate flow Y_1 will contain a transformed version of X. The transformation is done by the mix network. That is,

$$Y_1 = MixNetwork(X) + \text{noise packet counts},$$

where *noise packet counts* is caused by cross traffic. Intuitively, Y_1 has a stronger correlation with X than any other possible aggregate flow.

Theorem 1. *For a mix with any number of output links, the detection rate, D, is given by*

$$D \approx 1 - \sqrt{\frac{C_1}{N}} \times \int_{-\infty}^{-I_1\sqrt{\frac{N}{C_1}}} N(0,1)dx, \tag{9}$$

where N is the sample size, I_1 is the actual mutual information between Alice's flow packet count and the first aggregate flow packet count, $N(0,1)$ is the density function of the standard normal distribution, and C_1 is a constant defined in Equation (7).

Proof. We know that $\hat{I}_j$ satisfies a normal distribution. Its mean and variance can be derived from Lemma 1 and Lemma 2, respectively. Without loss of generality, we assume the first outgoing link contains Alice's flow. The mutual information estimation $\hat{I}_1$ between Alice's flow packet count X and the first aggregate flow packet count Y_1 has the following normal distribution:

$$\hat{I}_1 \sim N\left(I_1 + \frac{(r-1)(s-1)}{2N}, \frac{C_1}{N}\right). \tag{10}$$

Since the first aggregate flow contains a mix-network-

transformed version of Alice's flow, it is easy to see

$$\begin{aligned} C_1 &= \sum_{u,v_1} p(u,v_1)\left(\log\frac{p(u,v_1)}{p(u)p(v_1)}\right)^2 \\ &\quad -\left(\sum_{u,v_1} p(u,v_1)\log\frac{p(u,v_1)}{p(u)p(v_1)}\right)^2 \\ &\neq 0, \end{aligned}$$

where $p(u,v_1)$ refers to the joint distribution of (X,Y_1).

The mutual information $\hat{I}_j$ $(j>1)$ between Alice's flow packet count X and the j^{th} possible aggregate flow packet count Y_j has the following normal distribution:

$$\hat{I}_j \sim N\left(I_j + \frac{(r-1)(s-1)}{2N}, \frac{C_j}{N}\right),$$

where C_j is defined in Equation (7).

If we assume that Alice's flow packet count is approximately independent of flow packet counts of possible aggregate flow other than the first one, it is easy to see that $C_j = 0$, and $I_j = 0$. That is, the mutual information estimation $\hat{I}_j$ $(i \neq 1)$ degenerates into a constant $\frac{(r-1)(s-1)}{2N}$.

If we assume that the sample size N is sufficiently large and the mix's links have the same bandwidth, the detection rate Formula (3) becomes

$$\begin{aligned} D &= Pr\left(\hat{I}_1 > \frac{(r-1)(s-1)}{2N}, \cdots, \hat{I}_1 > \frac{(r-1)(s-1)}{2N}\right) \\ &= Pr\left(\hat{I}_1 > \frac{(r-1)(s-1)}{2N}\right). \end{aligned}$$

Since $I(X,Y_1)$ has a normal distribution as in Equation (10), we can easily obtain the detection rate D:

$$\begin{aligned} D &= \int_{\frac{(r-1)(s-1)}{2N}}^{+\infty} N\left(I_1 + \frac{(r-1)(s-1)}{2N}, \frac{C_1}{N}\right)dx \\ &= 1 - \int_{-\infty}^{\frac{(r-1)(s-1)}{2N}} N\left(I_1 + \frac{(r-1)(s-1)}{2N}, \frac{C_1}{N}\right)dx. \end{aligned} \quad (11)$$

After some transformations, Equation (11) becomes

$$D \approx 1 - \sqrt{\frac{C_1}{N}} \times \int_{-\infty}^{-I(X,Y_1)\sqrt{\frac{N}{C_1}}} N(0,1)dx.$$

□

Two observations are in place regarding Theorem 1: First, Formula (9) is applicable to a mix using any mix strategy (simple proxy without any batching and reordering, timed mix, continuous-time mix and others [11]) and accommodates any input flow types. This demonstrates the generality of Theorem 1. Second, the detection rate is an increasing function of the sample size N. This is consistent with common sense: any mix network will fail to maintain anonymity if the adversary is allowed to make an arbitrarily long observation.

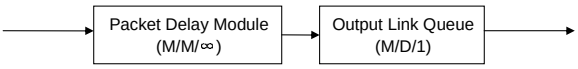


Figure 2: Model of a continuous-time mix

5.4 Joint Distribution of (X, Y_i) for a Continuous Mix

The Formula (9) in Theorem 1 is *generic* in terms of traffic and mix characteristics. However, both the constants C_i and the original mutual information I_i depend on the joint distribution function $p(u,v_i)$, which in turn depends on the traffic and the mix characteristics.

It has been shown earlier [5] that a continuous mix can be easily modelled as a two queue model shown in Figure 2, for the case of Poisson traffic. While it is generally agree upon that Poisson is a poor model for Internet traffic, it is adequate in our case, since we are interested in worst-case detectability of TCP flows. As illustrated in Figure 4, Poisson traffic is harder to detect, and therefore gives an safe lower bound on the detection rate.

The first queue of the continuous-time mix represents the packet delay module, while the second queue represents the fixed capacity output link of the mix. Since packets are delayed according to an exponential distribution, the delay module can be modelled as a $M/M/\infty$ queue. The output traffic of this queue is still a Poisson process, and, since packets in a mix network are typically padded to a fixed size, the output link queue can be modelled as $M/D/1$ queue.

Based on this model of a continuous-time mix, it is straightforward to derive the joint distribution of (X,Y_i) if we can model the incoming traffic into the mix.

Denote Y_i' as the packet count of the delay module's output flow, and Y_i' is also the packet count of the input flow to the i^{th} output link queuing module. Thus $X \to Y_i' \to Y_i$ forms a Markov chain. So the joint probability of (X,Y_i) is

$$\begin{aligned} & p(X=u, Y_i=v_i) \\ =& \sum_{v_i'=0}^{\infty} p(X=u, Y_i'=v_i', Y_i=v_i) \\ =& \sum_{v_i'=0}^{\infty} p(X=u)\cdot p(Y_i'=v_i'|X=u)\cdot p(Y_i=v_i|Y_i'=v_i'). \end{aligned} \quad (12)$$

According to our assumption about traffic arrival, the first term $p(X=u)$ in Equation (12) follows a Poisson distribution. The second term $p(Y_i'=v_i'|X=u)$ is determined by the packet delay module and the third term $p(Y_i=v_i|Y_i'=v_i')$ is determined by the output link queuing module.

Derivation of $p(Y_i'=v_i'|X=u)$ based on $M/M/\infty$ queuing:
Without loss of generality, we assume that Y_1' represents the packet count of the the first aggregate flow and

contains the transformed version of Alice's flow. Below, we first derive $p(Y_1' = v_1'|X = u)$ for the first aggregate flow and then $p(Y_i' = v_i'|X = u)$ $(2 \leq i \leq M)$ for the other aggregate flows.

A. Derivation of $p(Y_1' = v_1'|X = u)$*:*
Three sources of packets contribute to Y_1', the number of packet leaving the packet delay module during the sampling interval: (1) packets left over from the previous sampling interval, denoted as n_q, (2) Alice's packets arriving in the current sampling interval, denoted as n_f, and (3) noise packets arriving during the current sampling interval, denoted as n_z. Thus,

$$p(Y_1' = v_1'|X = u) = \sum_{n_q+n_f+n_z=v_1'} p(N_q = n_q) \, p(N_f = n_f|X = u)p(N_z = n_z). \quad (13)$$

The derivation of the three terms in Equation (13) can found in Appendix.

B. Derivation of $p(Y_i' = v_i'|X = u)$ *for* $i > 1$*:*
Since Alice's traffic is independent from traffic on the other links, easily we have $p(Y_i' = v_i'|X = u) = p(Y_i' = v_i')$. We can derive the probability $p(Y_i' = v)$ in the same way of deriving $p(N_z = n_z)$. We use $\lambda_{Y_i'}$ to denote the average rate of the traffic on the output link i $(i > 1)$:

$$p(Y_i' = v_i') = \sum_{z=v_i'}^{\infty} \sum_{|S_d|=v_i'} p_z(\lambda_{Y_i'}, S_d).$$

Derivation of $p(Y_i = v_i|Y_i' = v_i')$ based on $M/D/1$ queuing:
Similar to the above, we differentiate the case of $p(Y_1 = v_1|Y_1' = v_1')$ and $p(Y_i = v_i|Y_i' = v_i')$ where $i > 1$.

A. Derivation of $p(Y_1 = v_1|Y_1' = v_1')$*:*
The probability $p(Y_1 = v_1|Y_1' = v_1')$ is determined by the $M/D/1$ queue. We use Q_1 to denote the size of the queue at output Port 1. So the probability $p(Y_1 = v_1|Y_1' = v_1')$ can be expressed as follows:

$$p(Y_1 = v_1|Y_1' = v_1') = p(Q_1 = v_1 - v_1'), \quad (14)$$

when $v_1 < BW_1 \cdot T$, where in this subsection, BW_1 is the bandwidth of the first link. Obviously, when $v_1 < BW_1 \cdot T$, the probability $p(Y_1 = v_1|Y_1' = v_1')$ is zero if $v_1' > v_1$. Because $v_1 < BW_1 \cdot T$ means the link bandwidth is not fully utilized, the queue size will be zero. So all the v_1' incoming packets should depart in the sample interval. When $v_1 = BW_1 \cdot T$, we have

$$p(Y_1 = v_1|Y_1' = v_1') = p(Q_1 > BW_1 \cdot T - v_1') = \sum_{q=BW_1 \cdot T - v_1'}^{\infty} p(Q_1 = q). \quad (15)$$

The equilibrium state queue length distribution of the M/D/1 queue will be $p(Q_1 = 0) = 1 - \rho$. where $\rho = \frac{\lambda_z+\lambda_f}{BW_1}$, λ_z is the average rate of noise traffic on the first link, and λ_f is the average rate of Alice's flow. Similarly, $p(Q_1 = 1) = (1 - \rho)(e^{\rho} - 1)$, and

$$p(Q_1 = q) = (1-\rho)\sum_{j=1}^{q}(-1)^{q-j}[\frac{(j\rho)^{q-j}}{(q-j)!} + (1-\delta_{qj})\frac{(j\rho)^{q-j-1}}{(q-j-1)!}]e^{j\rho}$$

where $q \geq 2$ and $\delta_{qj} = \{^{1,q=j}_{0,q\neq j}$.

B. Derivation of $p(Y_i = v_i|Y_i' = v_i')$*,* $i > 1$*:*
The probability $p(Y_i = v_i|Y_i' = v_i')$ can be derived in the same way as in Equations (14) and (15).

In summary, by combining results derived above, we can obtain joint distributions for continuous mix.

6 Performance Evaluation

The evaluation of our work consists of three steps: first, we illustrate the validity of our theory through an experimental comparison of detection rates of Poisson vs. TCP traffic. Then we show results of ns-2 based simulations that illustrate the accuracy of our continuous-time mix model. Finally, we discuss the effects of the delay parameters of the continuous mix on the mix's effectiveness.

This series of experiments evaluates the accuracy of the model described in the paper for the case of Poisson traffic and a single mix. We show that the Poisson assumption is an acceptable one for this case, since it provides a conservative bound on the effectiveness of the flow correlation attack on TCP flows. Similarly, attacks on large-scale mix networks rely either on concatenations of single-mix attacks, or attacks on clusters of mixes (so-called super mixes), for which our model is accurate as well.

6.1 Network Setup in a Test-bed

The experimental test-bed and simulation network setup is shown in Figure 3. Alice sends traffic to Bob, while senders S_2 and S_3 send noise traffic to *both* Bob and R_2.

We assume that the adversary is interested in finding out whether Bob is the receiver of Alice's traffic. Since more complicated cases just require more comparisons, this setup is sufficient for us to demonstrate the accuracy of our model.

6.2 Failure of the Continuous-time Mix

Experimental Results: We first show the failure of the continuous-time mix by experiments. We implemented the continuous-time mix on the Timesys/RealTime Linux operating system. The mix control module that performs the delay function is integrated into the Linux firewall system using Netfilter. The bandwidth of all links is $10Mb/s$. The average delay of the continuous-time mix in this subsection is 20ms.

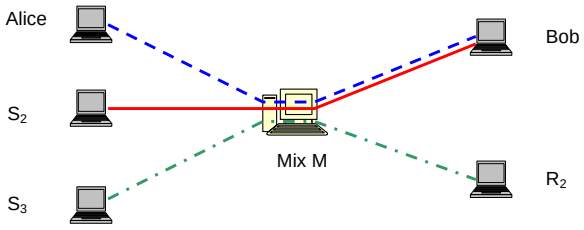


Figure 3: Experiment setup

We consider two cases here: (1) All the traffic is TCP. TTCP is used to generate TCP traffic. There are five TCP flows to Receiver Bob and R_2 respectively. One of the flow to Bob is from Alice; (2) All the traffic is Poisson (using UDP). The rate of traffic to Bob and to R_2 is around 650 packets/s and the rate of traffic from Alice to Bob is around 200 packets/s.

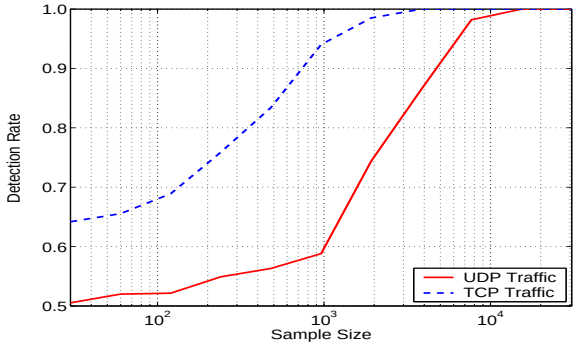


Figure 4: Detection rate of flow correlation attack under different network setting

The result of the flow correlation attacks on the continuous-time mix in the test-bed is shown in Figure 4. We make three observations:

1) For the continuous-time mix, flow correlation attacks can achieve high detection rates given access to sufficient data. Detection rates increase with the amount of data available. This result and previous experimental results in [15] empirically give evidence for the correctness of our detection rate Formula (9).

2) Experiments with TCP flows show much higher detection rates than experiments with UDP traffic. The reason for this is that TCP's feedback congestion control mechanism causes Alice's flow to have a stronger timing signature in comparison with the Poisson traffic. This signature is observable both at the input and the output of the mix. It is therefore easier to correlate Alice's flow with aggregates at the output links. Thus it is easier to detect than Poisson traffic.

3) Flow correlation attacks can be very efficient. Recall that we use a sampling interval of 10ms. Thus, a sample size of 3000 corresponds to a sample length of 30 seconds. Given access to 30 seconds of data, an attacker can achieve a detection rate of 100% in the case of TCP traffic and a detection rate of around 90% for Poisson traffic even with a high load of noise traffic.

Modelling Accuracy by Simulation: We use the ns-2 simulator to evaluate the accuracy of the model described in Section 5. We consider two cases of traffic load: *light traffic load* and *heavy traffic load*. We distinguish the two cases to assess the accuracy of the $M/D/1$-based model for the output port, as in the case of light traffic load, the second queue can be largely ignored. In the experiments, we vary the link capacity instead of the traffic load, with a 1Tb/s and 5Mb/s capacity for the light and heavy load respectively. The traffic of Alice's flow is Poisson with an average rate of 100 packet/s. The noise traffic to Receiver Bob and R_2 are also Poisson with average rates 400 and 500 packet/s respectively. The link delay between the mix and the receivers is 50ms. The links between senders and mix have 100Mb/s bandwidth and 1ms delay. The continuous-time mix's average delay is set to 20ms.

Figure 5 compares the results obtained from our model and by simulation. We make two observations:

1) The results from the model well match the simulation results. For example, the mean estimation error is only around 5% and the estimation error never exceeds 15%.

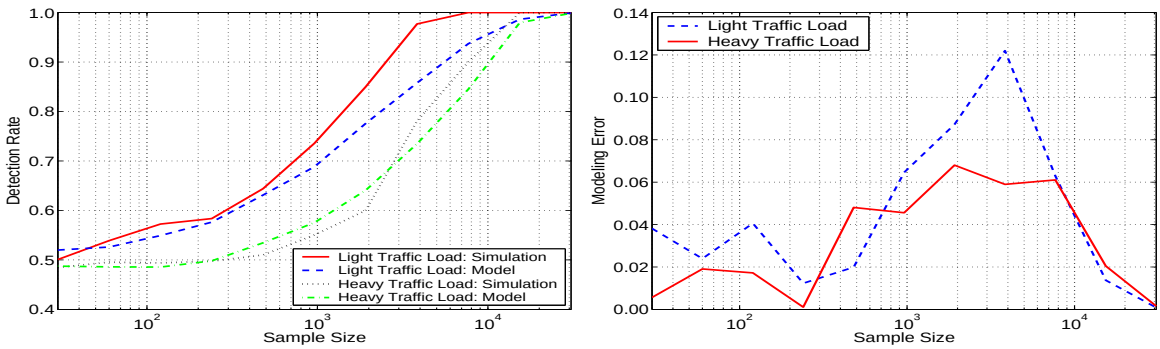
2) The detection rate is higher in the case of light traffic load. The reason is: in the case of heavy traffic load, the aggregate traffic rate is comparable to the link bandwidth. The output queue will therefore shape and so further perturb the outgoing traffic. This reduces the dependence between the sender's outbound flow and the receiver's inbound flow. Nevertheless, this effect is accurately captured by the $M/D/1$ queue model in this paper.

6.3 Impact of Continuous Mix Parameter

The continuous-time mix with exponentially distributed delay has a single parameter: the average delay t_{avg}. Figure (6) shows the relationship between the detection rate and the average delay for sample size 60, 480, 3840, and 30720. The sampling interval is set to 10ms. These sample size correspond to sample length of 0.6s, 4.8s, 38.4s and 307.2s. We make two observations:

1) Detection rate decreases as t_{avg} increases for each case of sample size. This is to be expected: because when t_{avg} increases, the probability for a packet held in the delay module or an incoming packet to leave the mix in the same sample interval will decrease. In turn, this will cause a smaller dependence between the flow of interest and the aggregate traffic containing the flow.

2) Detection rate increases as the sample size increases when we fix t_{avg}. This is consistent with the results



(a) Detection Rate by Simulation and Theory

(b) Mix Network without Batching

Figure 5: Modelling error

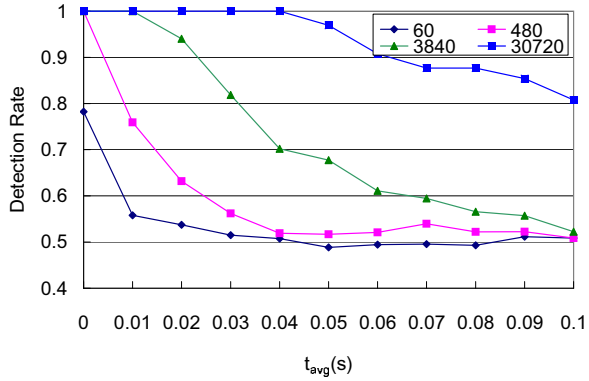


Figure 6: Impact of average mix delay parameter t_{avg}

in Figure 5. Again, it is because the increase of the amount of data for detection will cause more accurate estimation of dependency between the flow of interest and the aggregate traffic flows.

7 Conclusion

In this paper we formally model flow correlation attacks, which may severely degrade anonymous communication systems. Specially, we use queuing models to analyze the performance of a continuous-time mix, which randomly assign a deadline to each incoming packet.

While the effectiveness of flow correlation attacks was known and empirically demonstrated, we analytically model the relationship between the amount of information available to attackers and the detection rate. We define the latter as the probability that an adversary correctly determine the outgoing link taken by Alice's flow. Our formulae clearly show how an anonymity network ultimately fails under flow correlation attacks. By test-bed experiments and ns-2 simulations, we show the accuracy of our model and its use for designers to develop and deploy anonymity systems.

Our future work is to gain further understanding of the effectiveness of mix networks with TCP traffic. TCP traffic poses two challenges for mix network design. First, the indiscriminate delaying and possible reordering of packets in mix networks degrades TCP's goodput. Second, delays and reordering in mixes trigger second-order effects in TCP, such as congestion control, which in turn can negatively affect the level of anonymity provided by the mix network. Ultimately, mixes must be made TCP friendly, and so allow for high-performance anonymous communication for widely available application.

References

[1] A. Backm U. Möller and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity providing systems," in *Proceedings of Information Hiding Workshop (IH'01)*, pp. 245-257, Pittsburgh, PA, USA, 2001.

[2] P. Boucher, A. Shostack, and I. Goldberg, *Freedom Systems 2.0 Architecture*, White Paper, 2000.

[3] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of ACM*, vol. 24, no. 2, pp. 84-90, 1981.

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 1991.

[5] G. Danezis, "The traffic analysis of continuous-time mixes," in *Proceedings of Privacy Enhancing Technologies workshop (PET'04)*, pp. 35-50, Toronto, Canada, 2004.

[6] M. J. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02)*, pp. 193-206, Washington, DC, USA, 2002.

[7] C. Gülcü and G. Tsudik, "Mixing e-mail with babel," in *Proceedings of the Network and Distributed Security Symposium*, pp. 2-16, San Diego, CA, USA, 1996.

[8] D. Kesdogan, J. Egner, and R. Büschkes, "Stop-and-go mixes: Providing probabilistic anonymity in an open system," in *Proceedings of Information Hiding Workshop (IH'98)*, pp. 83-98, Portland, OR, USA, 1998.

[9] B. N. Levine, M. K. Reiter, C. Wang, and M. K. Wright, "Timing attacks in low-latency mix-based systems," in *Proceedings of Financial Cryptography*, pp. 251-265, Key West, FL, USA, 2004.

[10] M. Reiter and A. Rubin, "Crowds: anonymity for Web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66-92, 1998.

[11] A. Serjantov, R. Dingledine, and P. Syverson, "From a trickle to a flood: Active attacks on several mix types," in *Proceedings of Information Hiding Workshop (IH'02)*, pp. 36-52, Noordwijkerhout, The Netherlands, 2002.

[12] A. Serjantov and P. Sewell, "Passive attack analysis for connection-based anonymity systems," in *Proceedings of 8th European Symposium on Research in Computer Security*, pp. 116-131, Gjøvik, Norway, 2003.

[13] Q. Sun, D. R. Simon, Y. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, "Statistical identification of encrypted Web browsing traffic," in *IEEE Symposium on Security and Privacy*, pp. 19-30, Oakland, CA, USA, 2002.

[14] P. F. Syverson, D. M. Goldschlag, and M. G. Reed, "Anonymous connections and onion routing," in *IEEE Symposium on Security and Privacy*, pp. 44-54, Oakland, CA, USA, 1997.

[15] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "On flow correlation attacks and countermeasures in mix networks," in *Proceedings of Privacy Enhancing Technologies workshop (PET'04)*, pp. 207-225, Toronto, Canada, 2004.

Appendix A

Derivation of $p(Y_i' = v_i'|X = u)$ based on $M/M/\infty$ queuing: Here we derive of the three terms in Equation (13).

A.1. $p(N_q = n_q)$:
Obviously,

$$p(N_q = n_q) = \sum_{q=n_q}^{\infty} p(Q = q) \cdot \binom{q}{n_q} p_{qdep}^{n_q} (1 - p_{qdep})^{q-n_q}, \tag{16}$$

where p_{qdep} denotes the probability of a packet delayed from a previous interval by delay module of the continuous mix being released during the sample interval, and $p(Q = q)$ denotes the probability of q packets held by the delay module.

Due to the memoryless property of the exponential distribution employed by the delay module, the distribution of remaining delay time after the beginning of a sample interval still follows an exponential distribution with the same parameter. If we assume that the delay module uses an exponential distribution with parameter λ_d,

$$p_{qdep} = \int_0^T \lambda_d e^{-\lambda_d t} dt. \tag{17}$$

Since the system can be modelled as M/M/∞ queue, the distribution of queue size Q at the beginning of a sample interval is:

$$p(Q = q) = \frac{r^q e^{-r}}{q!}, \tag{18}$$

where $r = \frac{\lambda_f + \lambda_z}{C_1}$, λ_f and λ_z are the Poisson arrival rate for the flow from Alice and noise traffic coming in through the same port. Equation (18) holds because of the fact that the flow from Alice is independent of the other traffic through the same port and the sum of the two Poisson process is also a Poisson process with arrival rate $\lambda_f + \lambda_z$.

So from Equations (16), (17) and (18), we can compute the probability $p(N_q = n_q)$.

A.2. $p(N_f = n_f|X = u)$:
Clearly, when $u < n_f$, the probability $p(N_f = n_f|X = u)$ is zero because the number of packet departures from the flow from Alice in one sampling interval should be no greater than u, the packet arrivals of the flow. There are $\binom{u}{n_f}$ combinations of n_f departures from the u arrivals.

We first label the u incoming packets with sequence number from 1 to u. Suppose the n_f departures contain the packets with sequence number $d_1, d_2, \cdots, d_{n_f}$. We use S_d to denote the set of the sequence number. So $S_d = \{d_1, d_2, \cdots, d_{n_f}\}$.

Since the packet count arrival is Poisson distributed, the probability of exactly u arrivals in a sample interval T is

$$\begin{aligned} P(u) \quad = \quad & \int_{t_1=0}^{T} \lambda_f e^{-\lambda_f t_1} \cdot \int_{t_2=0}^{T-t_1} \lambda_f e^{-\lambda_f t_2} \cdots \\ & \int_{t_u=0}^{T-\sum_{i=1}^{u-1} t_i} \lambda_f e^{-\lambda_f t_u} \cdot (1 - \\ & \int_{t_{u+1}=0}^{T-\sum_{i=1}^{u} t_i} \lambda_f e^{-\lambda_f t_{u+1}} dt_{u+1}) dt_u \cdots dt_1. \end{aligned}$$

Let $\Delta_i(t, t_H)$ be defined as follows:

$$\Delta_i(t, t_H) = \begin{cases} \lambda_f e^{-\lambda_f t} \cdot (1 - \int_{t'=0}^{t_H - t} \lambda_d e^{-\lambda_d t'} dt'), & \text{if } i \notin S_d \\ \lambda_f e^{-\lambda_f t} \cdot \int_{t'=0}^{t_H - t} \lambda_d e^{-\lambda_d t'} dt', & \text{if } i \in S_d. \end{cases}$$

The probability that the n_f packets in S_d are released by

the delay module of the continuous-time mix is then

$$p_u(\lambda_f, S_d) = \int_{t1=0}^{T} \Delta_1(t_1, T) \int_{t2=0}^{T-t_1} \Delta_2(t_2, T-t_1) \cdots \int_{t_u=0}^{T-\sum_{i=1}^{u-1} t_i} \Delta_u(t_u, T-\sum_{i=1}^{u-1} t_i)(1 - \int_{t_{u+1}=0}^{T-\sum_{i=1}^{u} t_i} \lambda_f e^{-\lambda_f t_{u+1}} dt_{u+1}) dt_u dt_{u-1} \cdots dt_1. \quad (19)$$

For example, when $u = 4$ and $S_d = \{2, 4\}$, we can get

$$\begin{aligned} & p_4(\lambda_f, \{2,4\}) \\ = & \int_{t_1=0}^{T} \lambda_f e^{-\lambda_f t_1} \cdot (1 - \int_{t'_1=0}^{T-t_1} \lambda_d e^{-\lambda_d t'_1} dt'_1) \\ & \int_{t_2=0}^{T-t_1} \lambda_f e^{-\lambda_f t_2} \cdot \int_{t'_2=0}^{T-t_1-t_2} \lambda_d e^{-\lambda_d t'_2} dt'_2 \\ & \int_{t_3=0}^{T-t_1-t_2} \lambda_f e^{-\lambda_f t_3} \cdot (1 - \int_{t'_3=0}^{T-t_1-t_2-t_3} \lambda_d e^{-\lambda_d t'_3} dt'_3) \\ & \int_{t_4=0}^{T-t_1-t_2-t_3} \lambda_f e^{-\lambda_f t_4} \cdot \int_{t'_4=0}^{T-t_1-t_2-t_3-t_4} \lambda_d e^{-\lambda_d t'_4} dt'_4 \\ & (1 - \int_{t_5=0}^{T-t_1-t_2-t_3-t_4} \lambda_d e^{-\lambda_d t_5} dt_5) dt_4 dt_3 dt_2 dt_1. \end{aligned}$$

By summing up all the probabilities for the set of the same size, we can get

$$p(N_f = n_f | X = u) = \sum_{|S_d|=n_f} p_u(\lambda_f, S_d).$$

A.3. $p(N_z = n_z)$:
The probability $p(N_z = n_z)$ can be calculated in a similar way as the probability $p(N_f = n_f | X = u)$. For the same port noise traffic, we can get $p_z(\lambda_z, S_d)$ in a similar way deriving Equation (19), where λ_z denotes the traffic rate of the same port noise traffic.

Thus we can get

$$p(N_z = n_z) = \sum_{z=n_z}^{\infty} \sum_{|S_d|=n_z} p_z(\lambda_z, S_d).$$

B. Derivation of $p(Y'_i = v'_i | X = u)$ where $i > 1$:
Since Alice's traffic is independent from traffic of receivers other than Bob, easily we have

$$p(Y'_i = v'_i | X = u) = p(Y'_i = v'_i).$$

We can derive the probability $p(Y'_i = v)$ in the same way of deriving $p(N_z = n_z)$. We use λ_{Y_i} to denote the average rate of the traffic through Port 2.

$$p(Y'_i = v'_i) = \sum_{z=v'_i}^{\infty} \sum_{|S_d|=v'_i} p_z(\lambda_{Y'_i}, S_d).$$

Ye Zhu is an Assistant Professor in Department of Electrical and Computer Engineering at Cleveland State University. He was a research assistant in the NetCamo project and received his Ph. D. in Electrical and Computer Engineering Department from Texas A&M University. He received his B. Sc. in 1994 from Shanghai JiaoTong University and his M. Sc. in 2002 from Texas A&M University. His research interests include anonymous communication, network security, peer to peer networking and traffic engineering.

Xinwen Fu is an Assistant Professor in the College of Business and Information Systems at Dakota State University. He received his B. S. (1995) and M. S. (1998) in Electrical Engineering from Xi?an Jiaotong University, China and University of Science and Technology of China respectively. He obtained his Ph. D. (2005) in Computer Engineering from Texas A&M University. In 2005, he joined Dakota State University as a faculty member. His current research interests are in anonymity and other privacy issues in distributed systems such as the Internet, mobile and sensor networks. Xinwen Fu won the 2nd place in the graduate category of the International ACM student research contest in 2002 and the Graduate Student Research Excellence Award of the Department of Computer Science at Texas A&M University in 2004.

Riccardo Bettati received his Diploma in Informatics from the Swiss Federal Institute of Technology (ETH), Zuerich, Switzerland, in 1988 and his Ph.D. from the University of Illinois at Urbana-Champaign in 1994. From 1993 to 1995, he held a postdoctoral position at the International Computer Science Institute in Berkeley and at the University of California at Berkeley. He is currently Associate Professor in the Department of Computer Science at Texas A&M University. His research interests are in traffic analysis and privacy, real-time distributed systems, real-time communication, and network support for resilient distributed applications.