Sampled Traffic Analysis by Internet-Exchange-Level Adversaries

Steven J. Murdoch and Piotr Zieliński

University of Cambridge, Computer Laboratory
http://www.cl.cam.ac.uk/users/{sjm217, pz215}

Abstract. Existing low-latency anonymity networks are vulnerable to traffic analysis, so location diversity of nodes is essential to defend against attacks. Previous work has shown that simply ensuring geographical diversity of nodes does not resist, and in some cases exacerbates, the risk of traffic analysis by ISPs. Ensuring high autonomous-system (AS) diversity can resist this weakness. However, ISPs commonly connect to many other ISPs in a single location, known as an Internet eXchange (IX). This paper shows that IXes are a single point where traffic analysis can be performed. We examine to what extent this is true, through a case study of Tor nodes in the UK. Also, some IXes sample packets flowing through them for performance analysis reasons, and this data could be exploited to de-anonymize traffic. We then develop and evaluate Bayesian traffic analysis techniques capable of processing this sampled data.

1 Introduction

Anonymity networks may be split into two categories: high latency (e.g. Mixminion [1] and Mixmaster [2]) and low latency (e.g. Tor [3], JAP [4] and Freedom [5]). High latency networks may delay messages for several days [6] but are designed to resist very powerful attackers which are assumed to be capable of monitoring all communication links, so called *global passive adversaries*. However, the long potential delay makes these systems inappropriate for popular activities such as web-browsing, where low-latency is required. Although, in low-latency anonymity networks, communications are encrypted to maintain bitwise-unlinkability, timing patterns are hardly distorted, allowing an attacker to deploy traffic analysis to de-anonymize users [7,8,9]. While techniques to resist traffic analysis have been proposed, such as link padding [10], their cost is high and they have not been incorporated into deployed networks.

Instead, these systems have relied on the assumption that the global passive adversary is unrealistic, or at least those who are the target of such adversaries have larger problems than anonymous Internet access. But even excluding the global passive adversary, the possibility of partial adversaries remains reasonable. These attackers have the ability to monitor a portion of Internet traffic but not the entirety. Distributed low-latency anonymity systems, such as Tor, aim to resist this type of adversary by distributing nodes, in the hope that connections through the network will pass through enough administrative domains to prevent a single entity from tracking users.

N. Borisov and P. Golle (Eds.): PET 2007, LNCS 4776, pp. 167–183, 2007.

This raises the question of how to select paths through the anonymity network to maximize traffic analysis resistance. Section 2 discusses different topology models of the Internet and their impact on path selection. We suggest that existing models, based on *Autonomous System* (AS) diversity, do not properly take account of the fact that while, at the AS level abstraction, a path may have good administrative domain diversity, physically it could repeatedly pass through the same *Internet eXchange* (IX). Section 3 establishes, based on Internet topology measurements, to what extent the Tor anonymity network is vulnerable to traffic analysis at IXes.

Section 4 describes how IXes are particularly relevant since, to assist load management, they record traffic data from the packets being sent through them. As aggregate statistics are required and the cost of recording full traffic would be prohibitive, only sampled data is stored. Hence, the quality of data is substantially poorer than was envisaged during the design and evaluation of previous traffic analysis techniques. Section 5 shows that, despite low sampling rates, this data is adequate for de-anonymizing users of low-latency anonymity networks. Finally, Section 6 discusses further avenues of research under investigation.

2 Location Diversity in Anonymity Networks

Tor has been long suspected, and later confirmed [11,12], to be vulnerable to an attacker who could observe both the entry and exit point of a connection through an anonymity network. As no intentional latency is introduced, timing patterns propagate through the network and may be used to correlate input and output traffic, allowing an attacker to track connection endpoints.

Delaying messages, as done with email anonymity systems, would improve resistance to these attacks, at least for a small number of messages. However, the additional latency here (hours to days) would, if applied to web browsing, deter most users and so decrease anonymity for the remainder [13]. In addition to the scarce bandwidth in a volunteer network, full link-padding would also introduce catastrophic denial of service vulnerabilities, because all parties would need to stop communicating and re-negotiate flow levels when one party left. Hence, the only remaining defense against traffic analysis is to ensure that the adversary considered in the system threat model is not capable of simultaneously monitoring enough points in the network to break users' anonymity.

While this approach would be of no help against a global passive adversary, more realistic attackers' traffic monitoring capabilities are likely to be limited to particular jurisdiction(s), whether they derive from legal or extra-legal powers. This intuitively leads to the idea that paths through anonymity networks should be selected to go through as many different countries as possible. The hope here is that an attacker attempting to track connections might have the ability to monitor traffic in some countries, but not all those on the path.

Unfortunately, Feamster and Dingledine [14] showed this approach could actually hurt anonymity because international connections were likely to go through one of a very small number of *tier-1* Internet Service Providers (ISP) –

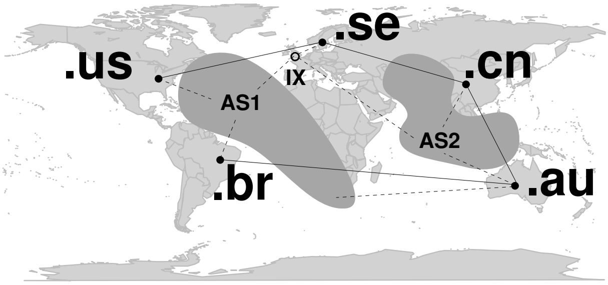


Fig. 1. Multiple-country path through a hypothetical anonymity network at geographical and AS level abstractions. Here, despite the path traveling through 3 countries between Brazil (.br) and the US (.us), there are two tier-1 ISPs which see all links. For example, the hop through China (.cn) is vulnerable since the incoming and outgoing links are observed by AS2. At first glance, the Swedish (.se) hop seems secure, as the incoming link is seen by AS2 and the outgoing by AS1. However, the Swedish ISP connects to AS1 and AS2 at LINX (IX), opening up the risk of observation there.

those who offer transit to the full Internet. Thus, connections to and from a far-flung Tor node are likely to both pass through a single tier-1 ISP, negating the anonymity benefit against an ISP level adversary. So, while – at the abstraction level of direct connections – a multi-country path may appear to have location diversity, by taking into account the ISPs that the data passes through between Tor nodes, weak points become clear, as shown in Fig. 1.

Instead, Feamster and Dingledine propose, when selecting paths, the relationship between ISPs carrying data between pairs of Tor nodes is taken into account. They did this by collecting Border Gateway Protocol (BGP) data, which controls how packets are routed between entities on the Internet, known as Autonomous Systems (AS) and roughly correspond to ISPs. From this data, assumptions about commercial relationships between ISPs, and heuristics about routing patterns, it is possible to estimate the ASes which will be on each path.

Optimizing path selection to maximize AS diversity reduces the likelihood that there will be one ISP who can observe the connection though the anonymity network at enough points to de-anonymize the user. However, although this level of abstraction is a substantial improvement over the naïve model of direct node connection, it does not fully take in account all potential monitoring points. This will be illustrated in the following section.

3 Impact of Internet Exchanges on Physical Topology

In the previous section, we discussed the advantages of selecting paths through anonymity networks such that there was no single AS which could monitor all hops between anonymity network nodes. This may be achieved by selecting nodes on ASes with high-degree i.e. those which are connected to multiple other ASes. ISPs owning such ASes might purchase cable connections to many other ISPs, but doing so would be extremely expensive. Instead, ISPs may connect their network to an IX, which will provide connectivity to all other ISPs with a presence at that IX. This approach is more prevalent in Europe than in the US, due to differing commercial structures and historical development; also because of language differences, intra-country traffic is substantial.

Thus, while at the AS level it appears that the path makes multiple transitions between distinct ASes, physically, each of these connections might pass through the same IX. Hence, despite the path attaining high AS diversity, there remains one entity who is able to de-anonymize the traffic. In order to establish how much of a problem this is for deployed anonymity networks, we set out to determine how successful an IX level adversary would be, compared to an AS level one, in de-anonymizing Tor users.

The techniques of Feamster and Dingledine [14] rely on building a map of AS paths from BGP data, but this is not helpful for our purposes as the IXes do not appear at this level. From the perspective of a router in an IX, packets travel directly to the destination AS. Furthermore, their approach depends on information about ISP relationships and routing policies which are a carefully guarded secret and so must be guessed. However, it is common practice to allocate each router in an IX an IP address from a single subnet.

Hence, while the AS path of a connection will not reveal whether it is going through an IX, a `traceroute` [15] is likely to. Unlike finding AS paths, collecting `traceroute` data requires access to the system at both ends of the path. As Tor does not currently implement a mechanism for performing `traceroute`s, the operator of the node must do so manually. To limit the effort to a feasible level, here we take the UK as a case study.

3.1 Experimental Results

Based on geo-location databases and manual investigation, we identified Tor nodes hosted in the UK and contacted the operators to request that they run a script to collect data to validate our hypothesis. One of our constraints was that no custom binary applications could be used, as the recipient could not easily confirm they were benign. Instead, we simply invoked the OS provided `traceroute` (or on Windows, `tracert`). These are not designed with speed or parallelism in mind, so to keep the runtime reasonable (2–24 hours, depending on timeouts) on the slower Windows test machines we only traced 140 destinations, and on *nix machines, tested 595 destinations. These destinations consisted of the same 15 websites and 11 US consumer ISPs tested in [14] and the remainder were randomly selected Tor nodes.

We received 19 (14 *nix, 5 Windows) responses from the 33 operators we were able to contact. This totaled 9 025 paths with an average path length of 14 hops (excluding failed responses). For each hop we established whether it was in one of the subnets of LINX (London InterNet eXchange), AMS-IX (AMSterdam Internet eXchange) or DE-CIX (the German Internet exchange, in Frankfurt). Also, using the Team Cymru Database [16], we established the BGP origin AS for each IP address. Note that although we are arranging data by AS, this path

Table 1. Number of paths passing through ASes and IXes.

AS name (ASN)	Paths	%
Level 3 (3356)	1 961	22%
NTL (5089)	1 445	16%
Zen (13037)	1 258	14%
JANET (786)	1 224	14%
Datahop (6908)	996	11%
Tiscali (3257)	953	11%
Sprint (1239)	935	10%
Cogent (174)	871	10%
Telewest (5462)	698	8%
Telia (1299)	697	8%

IX name (subnet)	Paths	%
LINX (195.66.224.0/22)	2 392	27%
DE-CIX (80.81.192.0/22)	231	3%
AMS-IX (195.69.144.0/22)	202	2%

is not the same as the BGP path discussed in [14]. Importantly, while IXes may have an AS, they do not broadcast routes, and so do not appear in BGP paths, whereas `traceroute` establishes the IP address of the border routers, from which the IX can be inferred.

The results are summarized in Table 1. As can be seen, Level 3, a large tier-1 ISP appears at least once on 22% of paths and other tier-1 ISPs, such as Tiscali, Sprint, Cogent and Telia also appear. Since our tests were all from UK Tor nodes, mainly run by volunteers, consumer ISPs also feature, such as NTL, Zen and Telewest, as does the UK academic network operator, JANET. Finally, Datahop, who provide connectivity between 10 data-centers in London, are present on 11% of paths. This broadly matches the results of [14], in that a small number of ISPs are present many paths.

However, if we now examine whether an IX is on the path, we find a new class of observation points. Despite being invisible at the BGP level, LINX is present on 27% of paths. There are 22 distinct ASes in the previous hop to LINX and 109 following the LINX hop, so AS-diverse paths will not substantially impact LINX crossings. Hence, exploiting the IX as an observation point is an effective attack against both existing and proposed anonymity network routing schemes. The connectivity graph of selected ASes, based on our data, is shown in Fig. 2.

4 Traffic Analysis from Sampled Data

The previous section has shown how an adversary positioned at an IX would be capable of monitoring a substantial quantity of traffic through the Tor network. A powerful adversary would be in a position to install expensive hardware to mount conventional traffic analysis attacks but such an adversary would likely be able to deploy other, more effective, attacks. However, the network infrastructure provided by an IX may already have the traffic analysis capabilities that a more modest attacker could use.

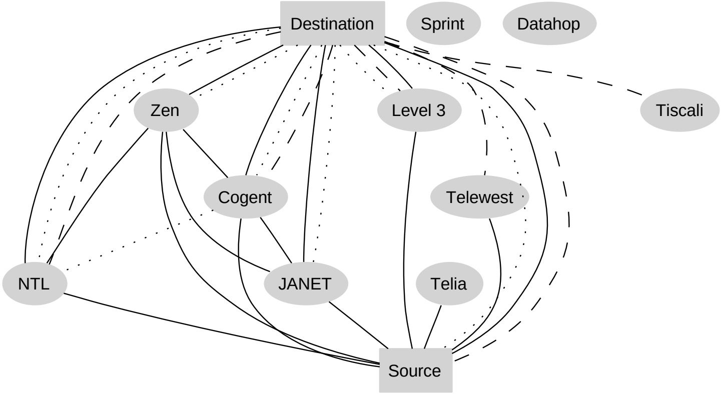


Fig. 2. AS connectivity via IX graph. Only ASes in Table 1 are shown and all sources and destinations are collapsed to single nodes. Links between ASes which pass through LINX are shown as solid lines, AMS-IX is shown by dotted lines and DE-CIX by dashed lines. Paths which go through none of these IXes are omitted. From this we can see that, in our data, connections through Sprint and Datahop go from source to destination without passing through any of the IXes we have selected.

To aid network management, high-end switches and routers have monitoring features which, although not designed for this purpose, may still be effective in tracing users of anonymity networks. This section will evaluate the suitability of network monitoring data for traffic analysis.

4.1 Traffic Monitoring in High-Speed Networks

On low-bandwidth small-office or business networks, full packet analysis tools such as `tcpdump` [17] are adequate to monitor traffic for debugging or to measure load. However, on links found on high-speed networks, the capacity required to store all packets rapidly becomes infeasible. For example, at time of writing, both LINX and AMS-IX carry approximately 150 Gb/s, which exceeds the theoretical maximum capacity of the high-speed PCIe bus, 64 Gb/s (32 lanes at 2 Gb/s each). Despite these difficulties, there is high demand for monitoring of such high-speed links, to detect problems such as routing loops, balance load across network infrastructure and anticipate future demands on capacity.

These applications do not rely on packet content, and for privacy reasons it may be desirable not to record this at all. Thus, medium to high-end networking equipment is commonly equipped with the ability to record aggregate data on the traffic passing through it. One such mechanism is *NetFlow* [18], developed by Cisco Systems but supported by other equipment manufacturers. NetFlow equipped infrastructure records unidirectional *flows* as defined by a tuple (source IP, destination IP, source port, destination port, IP protocol, interface, class of service). For each of these, the device will record information such as the number of packets, total byte count and bitwise-or of TCP flags.

A disadvantage of this approach is that it requires the network hardware to inspect every packet flowing through it. This can incur substantial load at higher network speeds, so to counter this difficulty *sampled NetFlow* only inspects a proportion q of packets. While sampling reduces CPU load, the network hardware must still store state for every flow it considers to be live, which could potentially be very large. An alternative, as adopted by *sFlow* [19], is to move the aggregation out of the network device by immediately exporting sampled packet headers. This approach also gives access to additional fields in packet headers, such as the sequence number, which could be useful for traffic analysis. However, to ensure generality, we will concentrate on information available in sampled NetFlow style data, which could be constructed from sFlow logs if needed (the converse is not true).

Not only is high-speed traffic monitoring possible with standard networking equipment, but it is common practice to do so. Two examples which are particularly relevant to this paper are that AMS-IX record data for traffic management monitoring [20] and LINX (who record 1 in 2 048 packets [21]) additionally are considering using sFlow data for detecting email spam [22]. The same data could also assist tracking users of an anonymity network because Section 3.1 showed that a significant number of Tor flows pass through an IX. In the following section we will examine how successful this type of traffic analysis would be.

4.2 Traffic Analysis Assumptions

There are two basic types of traffic analysis. The first treats the anonymity network as a “black-box” and only inspects traffic entering and leaving the network. The second approach additionally examines flows within the network, and so improves the accuracy of the attack. In this paper, we will concentrate on the former category. As this does not make any assumptions about the structure of the network, it is the more general approach. However, the techniques we present here could also be applied to the latter category of attacks, as intra-network Tor traffic will also often cross a small number of Internet exchanges.

We assume that the attacked flow passes through an attacker controlled IX on both its path into and out of the anonymity network. This would be the case if, for example, both the customer and site are hosted on ISPs whose backbone connection was through an IX under surveillance. Also, we assume that packet sampling is independently and identically distributed over the flow. Although some models of network hardware implement periodic sampling, rather than random, this assumption will remain true because Tor traffic makes up an insignificant proportion of overall traffic.

The attacker observes a single flow going into the network and wishes to establish which of several outgoing flows it corresponds to. This could be, for example, finding which website a known criminal is uploading stolen data to. Alternatively, the attacker might wish to discover who has uploaded a particular video to a news website – now there is one outgoing flow and many incoming candidates. In both cases, the attacker will have a number of candidates in mind who are also generating traffic at the same time, and for our simulation we assume that these produce around 1 000 flows per hour. We also assume that the adversary can distinguish Tor traffic from other traffic, which may trivially done by IP address and port number, based on information in the Tor directory.

5 Mathematical Analysis

5.1 Model

Our model consists of n client-server flows. Each flow $\boldsymbol{p} = p_1, \ldots, p_m$ is a collection of packets sent at times $t_1, \ldots, t_m$. We model the times as a Poisson process with a start time s, duration l, and rate r (average packets per second). These three parameters are chosen independently at random for each flow.

Neither s, l, r nor the flow $\boldsymbol{p}$ are directly observable. The attacker sees a down-sampled version of $\boldsymbol{p}$, in which each packet is retained independently with a fixed probability q, called a sampling rate (typically about 1/2 000). Each flow is sampled at the input and at the output, resulting in two vectors of times: $\boldsymbol{x}$ and $\boldsymbol{y}$. Given a flow $\boldsymbol{p}$, the sampling processes $\boldsymbol{p} \to \boldsymbol{x}$ and $\boldsymbol{p} \to \boldsymbol{y}$ are independent:

$$s, l, r \quad \overset{\text{Poisson}}{\longrightarrow} \quad \boldsymbol{p} \qquad\qquad \boldsymbol{x} \quad \overset{\text{sampling}}{\longleftarrow} \quad \boldsymbol{p} \quad \overset{\text{sampling}}{\longrightarrow} \quad \boldsymbol{y} \tag{1}$$

In an n-flow system, the attacker sees all n output vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, and one input vector $\boldsymbol{x}$, which corresponds to some $\boldsymbol{y}_k$. The task of the attacker is to compute the probability $P(T_k)$ that $\boldsymbol{x}$ corresponds to $\boldsymbol{y}_k$, for each k.

To simplify the model, we assume that no packet from $\boldsymbol{p}$ appears simultaneously in both $\boldsymbol{x}$ and $\boldsymbol{y}$. Since $\boldsymbol{x}$ and $\boldsymbol{y}$ are independently sampled from $\boldsymbol{p}$, a given packet from $\boldsymbol{p}$ appears in both $\boldsymbol{x}$ and $\boldsymbol{y}$ with the probability of $q^2 = 2.5 \cdot 10^{-7}$, that is, once every $1/q^2 = 4 \cdot 10^6$ packets ($\approx 2\,\text{GB}$). Seeing the same packet on the input and the output is thus very unlikely, which prevents packet-matching attacks [9] and makes independent random delays of individual packets practically unobservable in the sampled data. For simplicity, we therefore assume instantaneous packet transmission. Section 5.5 shows that introducing a moderate delay to the system does not change the effectiveness of our attack.

The assumption of no common packets in $\boldsymbol{x}$ and $\boldsymbol{y}$ allows us to simplify (1) by observing that $\boldsymbol{x}$ and $\boldsymbol{y}$ are now *independent* Poisson processes with rate rq.

$$\boldsymbol{x} \quad \overset{\text{Poisson}}{\longleftarrow} \quad s, l, rq \quad \overset{\text{Poisson}}{\longrightarrow} \quad \boldsymbol{y} \tag{2}$$

This simplification eliminates the original (unobservable) flow $\boldsymbol{p}$ from the model.

5.2 Basic Solution

Let T_k denote the event in which input $\boldsymbol{x}$ and output $\boldsymbol{y}_k$ belong to the same flow. In our model, the exact probabilities $P(T_k)$ can be uniquely determined from Bayes' formula:

$$P(T_k|\boldsymbol{y}_{1..n}, \boldsymbol{x}) = \frac{P(\boldsymbol{y}_{1..n}|T_k, \boldsymbol{x})P(T_k|\boldsymbol{x})}{\sum_i P(\boldsymbol{y}_{1..n}|T_i, \boldsymbol{x})P(T_i|\boldsymbol{x})}. \tag{3}$$

Probabilities $P(T_k|\boldsymbol{x})$ express our prior information about the target, possibly based on the sampled input flow $\boldsymbol{x}$ (but not output flow $\boldsymbol{y}$). For example, we might know that a particular server k is just more popular than others, or that it is the only one to regularly receive high-volume traffic and $\boldsymbol{x}$ looks to be high-volume. For simplicity, in the rest of the analysis, we treat all servers equally; any prior information can be easily taken into account using (3).

The probabilities $P(\boldsymbol{y}_{1..n}|\boldsymbol{x}, T_k)$ in (3) can be computed as follows:

$$P(\boldsymbol{y}_{1..n}|\boldsymbol{x}, T_k) = P(\boldsymbol{y}_k|\boldsymbol{x}, T_k) \prod_{i \neq k} P(\boldsymbol{y}_i) = \frac{P(\boldsymbol{y}_k|\boldsymbol{x}, T_k)}{P(\boldsymbol{y}_k)} \prod_i P(\boldsymbol{y}_i). \tag{4}$$

Here, we used the fact that output flows $\boldsymbol{y}_i$ are independent, and that $P(\boldsymbol{y}_i|T_k) = P(\boldsymbol{y}_i)$: the information about input-output connection T_k is only relevant for statements that involve both inputs and outputs (such as $P(\boldsymbol{y}_k, \boldsymbol{x}|T_k)$).

Since we are only interested in relative probabilities for different k's, we can ignore all factors independent of k, such as $P(\boldsymbol{x}|T_k) = P(\boldsymbol{x})$ or $\prod_i P(\boldsymbol{y}_i)$, as they would cancel out in (3) anyway:

$$P(T_k|\boldsymbol{y}_{1..n}, \boldsymbol{x}) \overset{(3)}{\sim} P(\boldsymbol{y}_{1..n}|\boldsymbol{x}, T_k) \overset{(4)}{\sim} \frac{P(\boldsymbol{y}_k|\boldsymbol{x}, T_k)}{P(\boldsymbol{y}_k)} = \frac{P(\boldsymbol{y}_k, \boldsymbol{x}|T_k)}{P(\boldsymbol{x}|T_k)P(\boldsymbol{y}_k)} \sim \frac{P(\boldsymbol{x}, \boldsymbol{y}_k|T_k)}{P(\boldsymbol{y}_k)}. \tag{5}$$

We therefore need to compute $P(\boldsymbol{y}_k)$ and $P(\boldsymbol{x}, \boldsymbol{y}_k|T_k)$. We are dealing with a single flow $\boldsymbol{x} \leftarrow \boldsymbol{p} \rightarrow \boldsymbol{y}_k$, so – to avoid notational clutter – we will drop the explicit index k and assumption T_k from our formulae. In the new notation, we have $P(\boldsymbol{y})$ and $P(\boldsymbol{x}, \boldsymbol{y})$, which can be computed from appropriate conditional probabilities by integrating out the unknown parameters s, l, r:

$$P(\boldsymbol{y}) = \int_{s,l,r} P(\boldsymbol{y}|s, l, r)P(s, l, r). \tag{6}$$

$$P(\boldsymbol{x}, \boldsymbol{y}) = \int_{s,l,r} P(\boldsymbol{x}, \boldsymbol{y}|s, l, r)P(s, l, r) = \int_{s,l,r} P(\boldsymbol{x}|s, l, r)P(\boldsymbol{y}|s, l, r)P(s, l, r). \tag{7}$$

The last equality holds because $\boldsymbol{x}$ and $\boldsymbol{y}$, generated by model (2), are independent given s, l, r. The distribution $P(s, l, r)$ expresses our prior knowledge about flow starting times, durations, and rates.

We divide the interval $[s, s + l]$ into infinitesimally small windows of size $\mathrm{d}t$. Since $\boldsymbol{y}$ is a Poisson process (2), the probability of observing a single packet in one such window is $rq\,\mathrm{d}t$. The probability of no packets in $[s, s+l]$ is e^{-rql}. Thus,

$$P(\boldsymbol{y}|s, l, r) = \begin{cases} e^{-rql}(rq\,\mathrm{d}t)^{n_{\boldsymbol{y}}} & \text{if all times in } \boldsymbol{y} \in [s, s + l], \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Here, $n_{\boldsymbol{y}}$ is the number of packets in $\boldsymbol{y}$. The same formula (with $n_{\boldsymbol{x}}$) holds for $P(\boldsymbol{x}|s, l, r)$. Since $P(\boldsymbol{x}, \boldsymbol{y}|s, l, r) = P(\boldsymbol{x}|s, l, r)P(\boldsymbol{y}|s, l, r)$, we also have

$$P(\boldsymbol{x}, \boldsymbol{y}|s, l, r) = \begin{cases} e^{-2rql}(rq\,\mathrm{d}t)^{n_{\boldsymbol{x}}+n_{\boldsymbol{y}}} & \text{if all times in } \boldsymbol{x}, \boldsymbol{y} \in [s, s + l], \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

5.3 Long-Lived Flows

We first consider a simplified model, in which all flows start at the same known time s and have the same known duration l (basically, $[s, s+l]$ is our observation window). The only factor distinguishing the flows is their (unknown) rate r. From (8), we get:

$$P(\boldsymbol{y}) = \int_r P(\boldsymbol{y}|r)P(r) = \int_r e^{-rql}(rq\,\mathrm{d}t)^{n_{\boldsymbol{y}}}P(r). \tag{10}$$

where $P(r)$ is our prior information about the rate r. Since r is a positive parameter, we express our complete lack of prior knowledge by using the scale-invariant Jeffrey's ignorance prior $P(r) \sim r^{-1}\,\mathrm{d}r$ [23]. This basically says that $\log r$ is distributed uniformly: the probability of $r \in [a, b]$ is proportional to $\log(b/a)$. For example, $r \in [1, 10]$ and $r \in [10, 100]$ have the same probability.

$$P(\boldsymbol{y}) \overset{(10)}{=} \int_r (rq\,\mathrm{d}t)^{n_{\boldsymbol{y}}} e^{-rql}P(r) = (q\,\mathrm{d}t)^{n_{\boldsymbol{y}}} \int_{r=0}^{\infty} r^{n_{\boldsymbol{y}}-1}e^{-rql}\,\mathrm{d}r = \frac{\mathrm{d}t^{n_{\boldsymbol{y}}}}{l^{n_{\boldsymbol{y}}}}\mathit{\Gamma}(n_{\boldsymbol{y}}). \tag{11}$$

We used $\int_0^\infty z^{a-1}e^{-bz}\,\mathrm{d}z = \mathit{\Gamma}(a)/b^a$; for integer n we have $\mathit{\Gamma}(n) = (n-1)!$.

Similarly, from (9),

$$P(\boldsymbol{x}, \boldsymbol{y}) = \int_r (rq\,\mathrm{d}t)^{n_{\boldsymbol{x}}+n_{\boldsymbol{y}}} e^{-2rql}P(r) = \frac{\mathrm{d}t^{n_{\boldsymbol{x}}+n_{\boldsymbol{y}}}}{(2l)^{n_{\boldsymbol{x}}+n_{\boldsymbol{y}}}}\mathit{\Gamma}(n_{\boldsymbol{x}}+n_{\boldsymbol{y}}). \tag{12}$$

We can now use (5) to compute the final probability:

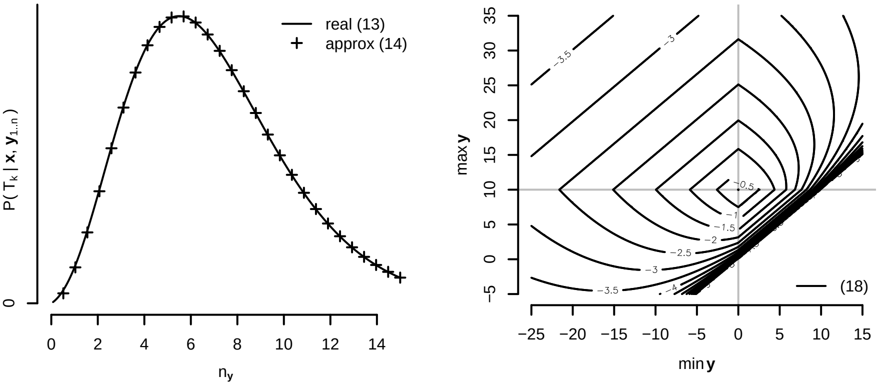
$$P(T_k|\boldsymbol{y}_{1..n}, \boldsymbol{x}) \sim \frac{P(\boldsymbol{x}, \boldsymbol{y}_k|T_k)}{P(\boldsymbol{y}_k)} = \frac{\mathrm{d}t^{n_{\boldsymbol{x}}}}{(2l)^{n_{\boldsymbol{x}}}} \cdot \frac{\mathit{\Gamma}(n_{\boldsymbol{x}}+n_{\boldsymbol{y}_k})}{2^{n_{\boldsymbol{y}_k}}\mathit{\Gamma}(n_{\boldsymbol{y}_k})} \sim \frac{\mathit{\Gamma}(n_{\boldsymbol{x}}+n_{\boldsymbol{y}_k})}{2^{n_{\boldsymbol{y}_k}}\mathit{\Gamma}(n_{\boldsymbol{y}_k})}. \tag{13}$$

Interpretation. Fig. 3(a) shows a normalized plot of (13) for $n_{\boldsymbol{x}} = 5$ as a function of $n_{\boldsymbol{y}}$. The maximum probability is assigned to $n_{\boldsymbol{y}} \approx n_{\boldsymbol{x}}$, when the numbers of observed packets on the input and on the output are similar. This confirms our intuition and also yields quantitative probabilities for different $n_{\boldsymbol{y}}$'s, which can be used for combining evidence from multiple observations.

The exact maximum occurs for $n_{\boldsymbol{y}} > n_{\boldsymbol{x}}$ because the prior $P(r) \sim r^{-1}\,\mathrm{d}r$ causes $P(r \in [4, 5]) > P(r \in [5, 6])$ (because $\frac{5}{4} > \frac{6}{5}$). This makes small $n_{\boldsymbol{y}}$'s more probable to be produced by chance than larger ones, decreasing their match probability. Using Stirling's approximation of $n!$, we get (see appendix):

$$P(T_k|\boldsymbol{y}_{1..n}, \boldsymbol{x}) \sim \frac{(n_{\boldsymbol{x}}+n_{\boldsymbol{y}}-1)^{n_{\boldsymbol{x}}+n_{\boldsymbol{y}}-\frac{1}{2}}}{2^{n_{\boldsymbol{y}}}(n_{\boldsymbol{y}}-1)^{n_{\boldsymbol{y}}-\frac{1}{2}}}, \tag{14}$$

which very closely matches the original, as shown in Fig. 3(a). The maximum of (14), obtained by comparing its derivative to zero, is $n_{\boldsymbol{y}} \approx n_{\boldsymbol{x}} + \frac{1}{2}$.



(a) $P(T_k|\boldsymbol{x}, \boldsymbol{y}_{1..n})$ given by (13) for fixed $n_{\boldsymbol{x}} = 5$ and $n_{\boldsymbol{y}}$ ranging from 0 to 15.

(b) $\log P(T_k|\boldsymbol{x}, \boldsymbol{y}_{1..n})$ given by (18) for $n_{\boldsymbol{x}} = n_{\boldsymbol{y}} = 5$, $\min \boldsymbol{x} = 0$, $\max \boldsymbol{x} = 10$, and variable $\min \boldsymbol{y}$ and $\max \boldsymbol{y}$.

Fig. 3. Relative probabilities based on (a) observed packet counts and (b) lengths.

5.4 General Flows

Now, we consider the general case, in which flows have different (unknown) durations l and starting times s. From (8), we can compute $P(\boldsymbol{y}|l, r)$ by integrating s out. For a given duration l, the possible starting times s belong to the interval $[\max \boldsymbol{y} - l, \min \boldsymbol{y}]$. If $l_{\boldsymbol{y}} = \max \boldsymbol{y} - \min \boldsymbol{y}$ is the observed length of $\boldsymbol{y}$, then this interval of possible values of s has the length $(l - l_{\boldsymbol{y}})_0 = \max\{l - l_{\boldsymbol{y}}, 0\}$. Assuming lack of prior knowledge about s (uniform prior $P(s) \sim \mathrm{d}s$), we have

$$P(\boldsymbol{y}|l, r) = \int_s P(\boldsymbol{y}|s, l, r)P(s) \overset{(8)}{\sim} (l - l_{\boldsymbol{y}})_0 e^{-rql}(rq\,\mathrm{d}t)^{n_{\boldsymbol{y}}}. \tag{15}$$

Using Jeffrey's priors $P(l) \sim l^{-1}\,\mathrm{d}l$ and $P(r) \sim r^{-1}\,\mathrm{d}r$, we get:

$$\begin{aligned} P(\boldsymbol{y}) = \int_{l,r} P(\boldsymbol{y}|l, r)P(l, r) = \int_{l,r} (l - l_{\boldsymbol{y}})_0 e^{-rql}(rq\,\mathrm{d}t)^{n_{\boldsymbol{y}}} l^{-1} r^{-1}\,\mathrm{d}r\,\mathrm{d}l = \\ (q\,\mathrm{d}t)^{n_{\boldsymbol{y}}} \int_l (l - l_{\boldsymbol{y}})_0 l^{-1} \int_r e^{-rql} r^{n_{\boldsymbol{y}}-1}\,\mathrm{d}r\,\mathrm{d}l = \\ (q\,\mathrm{d}t)^{n_{\boldsymbol{y}}} \int_l (l - l_{\boldsymbol{y}})_0 l^{-1} \Gamma(n_{\boldsymbol{y}})(ql)^{-n_{\boldsymbol{y}}}\,\mathrm{d}l = \\ \mathrm{d}t^{n_{\boldsymbol{y}}}\Gamma(n_{\boldsymbol{y}}) \int_{l=l_{\boldsymbol{y}}}^{\infty} (l - l_{\boldsymbol{y}}) l^{-n_{\boldsymbol{y}}-1}\,\mathrm{d}l = \mathrm{d}t^{n_{\boldsymbol{y}}}\Gamma(n_{\boldsymbol{y}}) \frac{l_{\boldsymbol{y}}^{-n_{\boldsymbol{y}}+1}}{n_{\boldsymbol{y}}(n_{\boldsymbol{y}} - 1)}. \end{aligned} \tag{16}$$

We can compute $P(\boldsymbol{x}, \boldsymbol{y})$ in a similar way. Let $n_{\boldsymbol{xy}} = n_{\boldsymbol{x}} + n_{\boldsymbol{y}}$ be the total number of packets in $\boldsymbol{x}$ and $\boldsymbol{y}$, and $l_{\boldsymbol{xy}} = \max\{\max \boldsymbol{x}, \max \boldsymbol{y}\} - \min\{\min \boldsymbol{x}, \min \boldsymbol{y}\}$

the observed length of superimposed sequences $\boldsymbol{x}$ and $\boldsymbol{y}$. In general, $l_{\boldsymbol{xy}} \neq l_{\boldsymbol{x}} + l_{\boldsymbol{y}}$.

$$P(\boldsymbol{x}, \boldsymbol{y}) = \int_{l,r} (l - l_{\boldsymbol{xy}})_0 e^{-2rql} (rq \,\mathrm{d}t)^{n_{\boldsymbol{xy}}} l^{-1} r^{-1} \,\mathrm{d}r \,\mathrm{d}l = \frac{\mathit{\Gamma}(n_{\boldsymbol{xy}}) \,\mathrm{d}t^{n_{\boldsymbol{xy}}}}{2^{n_{\boldsymbol{xy}}} (n_{\boldsymbol{xy}})(n_{\boldsymbol{xy}} - 1) l_{\boldsymbol{xy}}^{n_{\boldsymbol{xy}}-1}}. \tag{17}$$

Ignoring all factors independent of k, (5) gives us the final probability

$$P(T_k | \boldsymbol{x}, \boldsymbol{y}_{1..n}) = \frac{P(\boldsymbol{x}, \boldsymbol{y}_k | T_k)}{P(\boldsymbol{y}_k)} \sim \frac{\mathit{\Gamma}(n_{\boldsymbol{xy}_k})}{2^{n_{\boldsymbol{xy}_k}} \mathit{\Gamma}(n_{\boldsymbol{y}_k})} \cdot \frac{n_{\boldsymbol{y}_k}(n_{\boldsymbol{y}_k} - 1)}{n_{\boldsymbol{xy}_k}(n_{\boldsymbol{xy}_k} - 1)} \cdot \frac{l_{\boldsymbol{y}_k}^{n_{\boldsymbol{y}_k}-1}}{l_{\boldsymbol{xy}_k}^{n_{\boldsymbol{xy}_k}-1}}. \tag{18}$$

Interpretation. Formula (18) consists of three factors: (i) the rate formula (13), (ii) a rate-dependent correction $n_{\boldsymbol{y}}(n_{\boldsymbol{y}} - 1)/(n_{\boldsymbol{xy}}(n_{\boldsymbol{xy}} - 1))$, and (iii) the length-dependent factor $l_{\boldsymbol{y}}^{n_{\boldsymbol{y}}-1}/l_{\boldsymbol{xy}}^{n_{\boldsymbol{xy}}-1}$, which is of the most interest to us here.

Consider matching an input flow with the observed starting time $\min \boldsymbol{x} = 0$, ending time $\max \boldsymbol{x} = 10$, and $n_{\boldsymbol{x}} = 5$ observed packets, against output flows $\boldsymbol{y}$ with the same number of observed packets $n_{\boldsymbol{y}} = 5$. For various starting and ending times $\min \boldsymbol{y}$ and $\max \boldsymbol{y}$, Fig. 3(b) presents the matching likelihood assigned by (18) (since $n_{\boldsymbol{x}}$ and $n_{\boldsymbol{y}}$ are constant, so are the first two factors).

As expected, the maximum is attained when the observed starting and ending times of both flows coincide: $\min \boldsymbol{x} = \min \boldsymbol{y} = 0$ and $\max \boldsymbol{x} = \max \boldsymbol{y} = 10$. Each contour line consists of two parallel straight lines joined by two curves. The two straight lines correspond to the observed input flow period completely containing the observed output flow period, and vice versa.

Optimality. The derivation of (18) is strictly Bayesian, so – given the model assumptions – the result is exact and uses all relevant information. Note that, despite the timings of all packets being available through $\boldsymbol{x}$ and $\boldsymbol{y}$, formula (18) uses only the total packet counts ($n_{\boldsymbol{y}}$, $n_{\boldsymbol{xy}}$) and the observed lengths ($l_{\boldsymbol{y}}$, $l_{\boldsymbol{xy}}$). This shows that the exact timings of individual packets (used by timing-based attacks) are irrelevant for the inference in our model.

5.5 Evaluation

To evaluate the effectiveness of our method in attacking an individual Tor node, we first collected real traffic distributions of observed flow rates and durations (Fig. 4). Then, we performed a number of simulations of a 120 min execution of a node. Flow durations (1–30 min) and rates (0.1–50 packets/s) were drawn from the log-uniform ($P(z) \sim z^{-1} \,\mathrm{d}z$) prior, consistent with Fig. 4. Starting times were selected uniformly from the interval $[0, 120\,\mathrm{min} - l]$.

Our scoring method was “1” if the highest probability was assigned to the correct target, and “0” otherwise (if $i > 1$ targets shared the top probability,

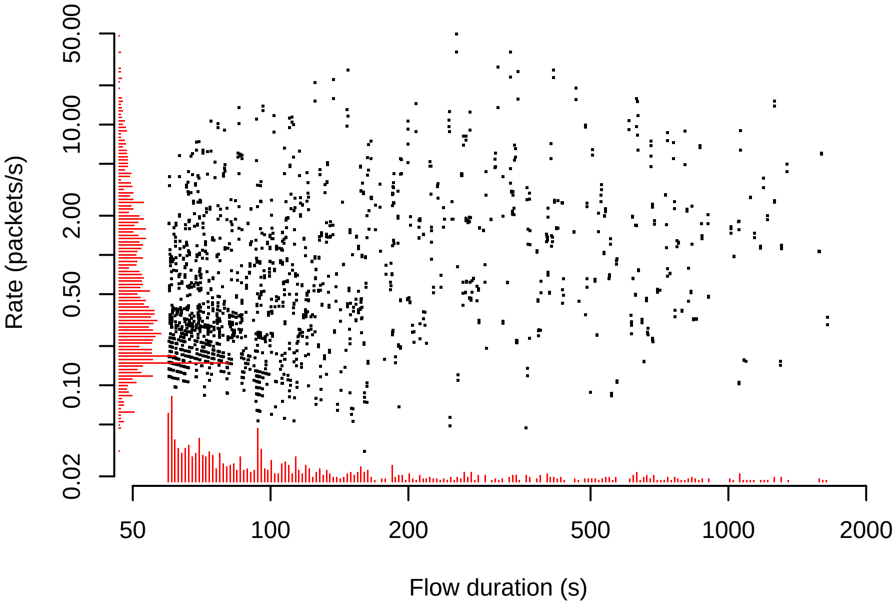


Fig. 4. Distribution of observed rates and flow durations on a single Tor node. Only flows that completed the three-way TCP handshake, at least 1 minute long, and consist of at least 5 packets are shown. Flows are closed after being idle for 1 minute.

then the score was $1/i$ instead of 1). For each simulation, we applied the attack independently to each input, and then averaged the results.

We varied the following parameters: the number of flows per hour (50–1 000), the sampling rate q (1/100–1/2 000), the mean network latency (0–10 min), and the attack method. Our parameter ranges are consistent with their real values: our Tor node transmitted 479 flows/h on average, the average Tor network latency was 0.5 s, and the current typical sampling rate is 1/2 048, but may increase in the future. The results of our simulations are summarized in Fig. 5.

Average number of flows. Fig. 5(a) confirms that more flows provide more protection. For a typical number of 500 flows/h, the attack had a 50% chance of success when the target sends ≈ 20 000 packets, that is ≈ 10 MB of data. With 50 flows/h, the same success rate required only 7 000 packets (3.5 MB).

Sampling rate. Fig. 5(b) suggests that the effectiveness of the attack depends only on the number of sampled packets, so doubling the sampling rate is equivalent to doubling the number of transmitted packets. For the technically feasible sampling rate of 1/100, a success rate of 50% required only 1 000 transmitted packets (500 kB).

Attack methods. We compared the following attacks: (i) *rate attack*, which applies (13), taking into account the observed number of packets and ignoring

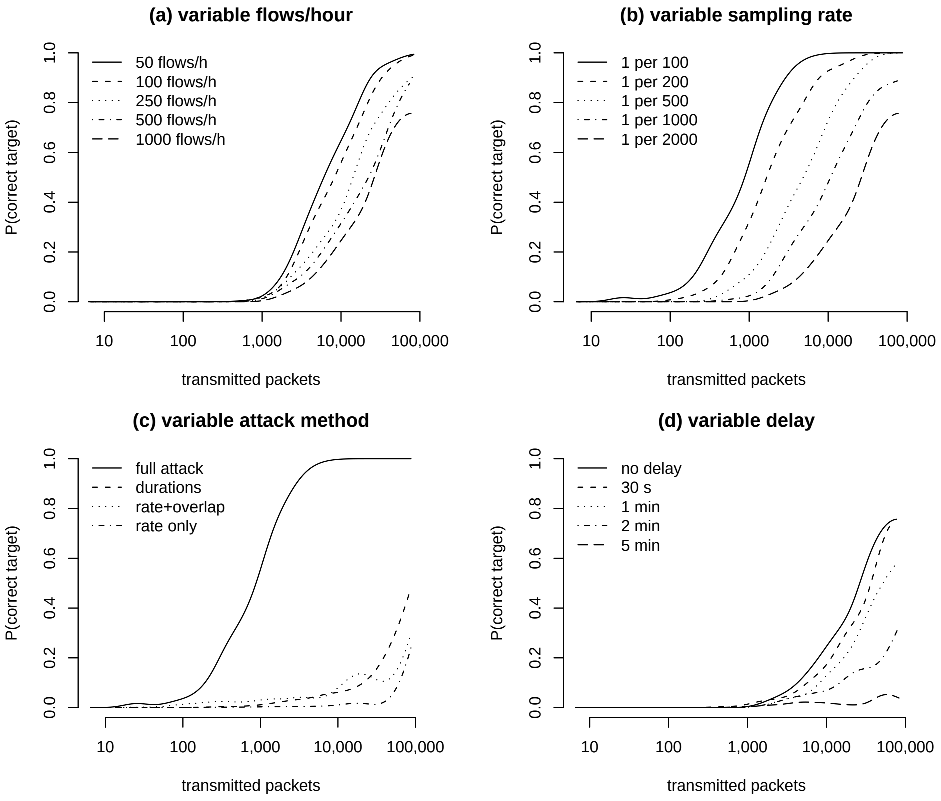


Fig. 5. Simulation results: the probability of choosing the correct target, as a function of the number of transmitted packets, for varying numbers of flows/hour (default 1 000), sampling rate (1/2 000, except (c)), attack method (full attack), and delay (0).

packet times; (ii) *rate+overlap attack*, which additionally ignores outputs with observed packet timings disjoint with the input; (iii) *length attack*, which selects the output $\boldsymbol{y}$ with the highest ratio $l_{\boldsymbol{y}}/l_{\boldsymbol{xy}}$; (iv) *full attack*, which uses (18).

Fig. 5(c) shows the effectiveness of these four methods in a system with a sampling rate of 1/100. The combined rate and length information (18) resulted in a 50% success rate for $\approx 1\,000$ packets (10 sampled). In comparison, taking only one factor (rate or length) into account, required 100 times more packets to achieve the same accuracy.

Delays. Fig. 5(d) shows the effects of introducing an exponentially distributed random delay to the system. The effectiveness of our attack stayed approximately the same for delays up to 30 s, and then started to deteriorate, reaching the 0% level for a 5 min delay. Note, however, that our attack explicitly assumes no delay whatsoever, therefore this result does *not* mean that a 5-minute random delay safeguards against all sampling attacks.

6 Future Work

For simplicity, we ignored several phenomena that occur in practice, such as different sampling rates and how Tor cells are split over IP packets. Generalizing our analysis to support different known sampling rates at input and output seems straightforward (but an attack by a single adversary with a fixed sampling rate is most likely). Similarly, the effect of packet splitting by Tor nodes seems to be statistically equivalent to different sampling rates. Our analysis could also be modified to take TCP sequence numbers, available from sFlow records, into account, to give more accurate rate calculation.

As reasonable random delays do not protect against our attack, we plan to examine other defenses, such as a moderate amount of dummy traffic. We would also like to measure the effectiveness of our attack against real systems, using an empirically determined prior distribution on durations and rates, for both the analysis (numerical integration required) and the evaluation. Ideally, such an evaluation should be performed for the entire Tor system, with its average 1 million flows per hour.

Furthermore, we are considering how intra-network traffic analysis could be performed. Similar techniques could be used, and are likely to work better than whole-network analysis since the number of flows will be smaller. However, there are complications which must be considered, in particular that multiple flows between the same pair of Tor nodes may be multiplexed within one encrypted TLS tunnel. An improved analysis would take this possibility into account and empirical studies would show to what extent this interferes with analysis.

7 Conclusion

We have demonstrated that Internet exchanges are a viable, and previously unexamined, monitoring point for traffic analysis purposes. They are present on many paths through our sample of the Tor network, even where BGP data would not detect any common points of failure. Furthermore, Internet exchanges are particularly relevant as in some cases they may record, and potentially retain data adequate to perform traffic analysis.

To validate to what extent this was true, we developed traffic analysis techniques which work on the sampled data which is being collected in practice by Internet exchanges. Using a Bayesian approach, we obtained the best possible inference, which means that we can not only attack vulnerable systems, but also declare others as safe under our threat model. Our probability formula is difficult to obtain by trial-and-error, and – as we show – can give orders of magnitude better results than simple intuitive schemes.

We also show that exact "internal" packet timings are irrelevant for optimum inference, so timing-based attacks cannot work with sparsely sampled data. For the same reason, deliberate random packet delays do not protect low-latency anonymity systems against our attack, as the minimum sensible latency (1 min) is unacceptable for web browsing and similar activities.

Acknowledgments We thank Richard Clayton, Chris Hall, Markus Kuhn, Andrei Serjantov and the anonymous reviewers for productive comments, and also the Tor node operators who collected the data used in this paper.

References

1. Danezis, G., Dingledine, R., Mathewson, N.: Mixminion: Design of a Type III Anonymous Remailer Protocol. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy. (2003)
2. Möller, U., Cottrell, L., Palfrader, P., Sassaman, L.: Mixmaster Protocol – Version 2. Draft (2003)
3. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium. (2004)
4. Berthold, O., Federrath, H., Köpsell, S.: Web MIXes: A system for anonymous and unobservable Internet access. In Federrath, H., ed.: Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, Springer-Verlag, LNCS 2009 (2000) 115–129
5. Boucher, P., Shostack, A., Goldberg, I.: Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc. (2000)
6. Serjantov, A., Murdoch, S.J.: Message splitting against the partial adversary. In: Proceedings of Privacy Enhancing Technologies workshop (PET 2005), Springer-Verlag, LNCS 3856 (2005)
7. Serjantov, A., Sewell, P.: Passive attack analysis for connection-based anonymity systems. In: Proceedings of ESORICS 2003. (2003)
8. Levine, B.N., Reiter, M.K., Wang, C., Wright, M.K.: Timing attacks in low-latency mix-based systems. In Juels, A., ed.: Proceedings of Financial Cryptography (FC '04), Springer-Verlag, LNCS 3110 (2004)
9. Danezis, G.: The traffic analysis of continuous-time mixes. In: Proceedings of Privacy Enhancing Technologies workshop (PET 2004). Volume 3424 of LNCS. (2004)
10. Dai, W.: Pipenet 1.1. Post to Cypherpunks mailing list (1998) `http://www.eskimo.com/~weidai/pipenet.txt`.
11. Øverlier, L., Syverson, P.: Locating hidden servers. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy, IEEE CS (2006)
12. Bauer, K., McCoy, D., Grunwald, D., Kohno, T., Sicker, D.: Low-resource routing attacks against anonymous systems. Technical Report CU-CS-1025-07, University of Colorado at Boulder (2007)
13. Acquisti, A., Dingledine, R., Syverson, P.: On the Economics of Anonymity. In Wright, R.N., ed.: Proceedings of Financial Cryptography (FC '03), Springer-Verlag, LNCS 2742 (2003)
14. Feamster, N., Dingledine, R.: Location diversity in anonymity networks. In: Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004), Washington, DC, USA (2004)
15. Jacobson, V.: traceroute (1) (1987) `ftp://ftp.ee.lbl.gov/traceroute.tar.gz`.
16. Team Cymru: IP to ASN lookup (v1.0) `http://asn.cymru.com/`.
17. Jacobson, V., Leres, C., McCanne, S.: tcpdump (1) (1989) `http://www.tcpdump.org/`.
18. Claise, B.: Cisco systems NetFlow services export version 9. RFC 3954, IETF (2004)

19. Phaal, P., Panchen, S., McKee, N.: InMon corporation's sFlow: A method for monitoring traffic in switched and routed networks. RFC 3176, IETF (2001)
20. Jasinska, E.: sFlow – I can feel your traffic. In: 23C3: 23rd Chaos Communication Congress. (2006) `http://events.ccc.de/congress/2006/Fahrplan/attachments/1137-sFlowPaper.pdf`.
21. Hughes, M.: LINX news. `http://www.uknof.org.uk/uknof4/Hughes-LINX.pdf` (2006)
22. Clayton, R.: spamHINTS project (2006) `http://www.spamhints.org/`.
23. Jaynes, E.T.: Probability Theory: The Logic of Science. Cambridge University Press (2003)

A Appendix

Theorem 1. *Formula* (13) *attains maximum for* $n_{\boldsymbol{y}} \approx n_{\boldsymbol{x}} + \frac{1}{2}$.

Proof. Stirling's factorial approximation gives us

$$n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n}.$$

Denoting $a = n_{\boldsymbol{x}}$, $b = n_{\boldsymbol{y}}$, and $c = a + b$, we have:

$$P(T_k|\boldsymbol{y}_{1..n}, \boldsymbol{x}) \sim \frac{\Gamma(a+b)}{2^b \Gamma(b)} = \frac{(c-1)!}{2^b (b-1)!} \approx \frac{\left(\frac{c-1}{e}\right)^{c-1} \sqrt{2\pi(c-1)}}{2^b \left(\frac{b-1}{e}\right)^{b-1} \sqrt{2\pi(b-1)}} \sim \frac{(c-1)^{c-\frac{1}{2}}}{2^b (b-1)^{b-\frac{1}{2}}} = X. \quad (19)$$

Instead of finding the maximum of X, it is easier to find the maximum of $\log X$:

$$\log X = (c - \tfrac{1}{2}) \log(c-1) - b \log 2 - (b - \tfrac{1}{2}) \log(b-1). \quad (20)$$

We can find the maximum of $\log X$ by differentiating it w.r.t. b, and remembering that $c' = (a+b)' = 1$:

$$\begin{aligned}(\log X)' &= \log(c-1) + \frac{c-\frac{1}{2}}{c-1} - \log 2 - \log(b-1) - \frac{b-\frac{1}{2}}{b-1} \\ &= \log(c-1) + \frac{1}{2(c-1)} - \log 2 - \log(b-1) - \frac{1}{2(b-1)} \\ &\approx \log(c - \tfrac{1}{2}) - \log 2 - \log(b - \tfrac{1}{2}) = \log\left(\frac{c-\frac{1}{2}}{2b-1}\right).\end{aligned} \quad (21)$$

Now, $(\log X)' = 0$ implies $c - \frac{1}{2} = 2b - 1$, which implies $b = a + \frac{1}{2}$, that is $n_{\boldsymbol{y}} = n_{\boldsymbol{x}} + \frac{1}{2}$.