Location Diversity in Anonymity Networks

Nick Feamster
MIT Computer Science and AI Laboratory
feamster@csail.mit.edu

Roger Dingledine
The Free Haven Project
arma@mit.edu

ABSTRACT

Anonymity networks have long relied on diversity of node location for protection against attacks—typically an adversary who can observe a larger fraction of the network can launch a more effective attack. We investigate the diversity of two deployed anonymity networks, Mixmaster and Tor, with respect to an adversary who controls a single Internet administrative domain.

Specifically, we implement a variant of a recently proposed technique that passively estimates the set of administrative domains (also known as autonomous systems, or ASes) between two arbitrary end-hosts without having access to either end of the path. Using this technique, we analyze the AS-level paths that are likely to be used in these anonymity networks. We find several cases in each network where multiple nodes are in the same administrative domain. Further, many paths between nodes, and between nodes and popular endpoints, traverse the same domain.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing Protocols

General Terms

Measurement, Security

Keywords

anonymity, mix networks, interdomain routing

1. INTRODUCTION

Anonymity networks aim to provide communications privacy for individuals or groups on the Internet, but these networks are still vulnerable to powerful eavesdroppers. A variety of organizations, ranging from corrupt law enforcement to curious ISPs, can passively observe large pieces of the Internet. Against high-latency *mix networks* such as Mixmaster [27], an adversary who observes a large volume of network traffic can notice over time that certain recipients are more likely to receive messages after particular senders have transmitted messages [15, 26]. Low-latency networks like Onion Routing [18, 31] are more directly vulnerable: an eavesdropper on both ends of the connection can quickly link sender to recipient through packet counting or timing attacks [16, 23, 35].

Permission to make digital or hard copies of all or part o f this work for personal or classroom use is granted without fee provided that c opies are not made or distributed for profit or commercial advantage and that co pies bear this notice and the full citation on the first page. To copy otherwis e, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WPES'04, October 28, 2004, Washington, DC, USA.
Copyright 2004 ACM 1-58113-968-3/04/0010 ...$5.00.

Anonymity designs use three strategies to mitigate these attacks.

- **Batching and pooling:** The network collects a group of input messages and reorders them before they exit, to hinder the adversary from learning which message in the batch originated from a given sender [12, 34].
- **Padding:** Senders provide decoy traffic as well as normal traffic to complicate the adversary's attempts to correlate sender and receiver [8, 14, 23].
- **Dispersal:** Reducing the chance that the adversary sees both endpoints for a given communication may entirely block some attacks on low-latency networks, and slow intersection attacks on high-latency networks.

Dispersal can be achieved by increasing the number of nodes in the network so an adversary of a given strength sees less of the network [1, 6, 33]; by arranging the overlay topology so messages can enter or exit at more places in the network (compared to a cascade topology [9]); and by *location diversity*—coordinating network behavior so each transaction is spread over multiple jurisdictions.

In this paper, we investigate a variant of location diversity that takes advantage of the fact that the Internet is divided into thousands of independently operated networks called *autonomous systems* (ASes). By considering the underlying topology of Internet routing, we can assess the vulnerability of existing mix networks to certain classes of adversary. Specifically, our *location independence* metric reflects the probability that the path to the entry point of a mix network and the path from the exit point will traverse the same AS. We then consider the topologies and node selection algorithms of two existing mix networks—Tor [18] and Mixmaster [27]—and evaluate the independence metric for these networks.

This paper presents several interesting results. First, we find that both Tor and Mixmaster have multiple nodes in the same autonomous system from different IP address spaces. Users of these networks should take care to avoid selecting two nodes from the same AS. In light of this, we argue that node selection algorithms that look only at IP prefixes, such as those used in Tarzan [19] and MorphMix [33], are likely to be less effective at achieving location independence.

Next, we measure the location independence of paths inside the mix network. We find that for short paths, given existing mix network topologies, the Mixmaster and Tor node selection algorithms will frequently create paths that can be observed by a single AS. Longer mix paths greatly reduce the likelihood that a single AS can observe a significant fraction of links in the path.

Finally, using a model of typical senders and receivers in anonymity networks, we measure the likelihood that a single AS can observe both the path from the initiator to the entry node and the path from the exit node to the responder; we find that entry and exit paths resulting from random node selection—even when the initiator never chooses the same node for both entry and exit—are likely to be observed by a single AS between 10% and 30% of the time, depending on the location of the initiator and responder, and that the single AS that can observe these paths is always a backbone ISP. We conclude that a slightly different node selection algorithm can allow users to minimize the likelihood that their entry path and exit path traverse the same AS.

2. BACKGROUND

We first describe the different types of mix networks and present a brief explanation of the types of attacks that each type of mix network must protect against. Then, we provide some background on Internet routing and topology.

2.1 Anonymity networks

Chaum [12] proposed hiding the correspondence between sender and recipient by wrapping messages in layers of public-key cryptography, and relaying them through a path composed of *mixes*. Each mix in turn decrypts, delays, and re-orders messages, before relaying them onward.

Subsequent anonymity systems have diverged in two directions. Systems like Babel [22], Mixmaster, and Mixminion [17] defend against powerful adversaries at the cost of requiring high and variable latency. Other systems, such as Onion Routing [31], its successor Tor [18], and the Freedom network [10], support low-latency transactions such as web browsing, but necessarily have a weaker threat model. Onion Routing and Freedom differ from single-hop proxies like the Anonymizer [3] or fixed-path topologies like Web Mixes [7] in that they aim to achieve as much diversity in node placement and path selection as possible.

Anonymity networks try to protect against a wide variety of both passive and active attacks [5, 30]. Such attacks generally fall into two categories: attacks inside the network and endpoint attacks. Attacks inside the network partition anonymity sets through passive observation [9, 17] or active traffic manipulation [34], or otherwise narrow the set of suspects for a given transaction. Endpoint attacks treat the network as a black box and consider only the entry node and exit node for the transaction; such attacks include simple timing and counting attacks against low-latency systems [23, 35] and long-term intersection or disclosure attacks against high-latency systems [9, 15, 26].

Mixmaster and Tor are deployed networks with dozens of nodes around the world (Appendix B lists the nodes in each network). We will describe their threat models in Section 3 and their path selection algorithms in Section 4.1.

Previous work has recognized the importance of location independence. Tim May and Eric Hughes wrote about the idea of location independence in early posts to the cypherpunks list. Mixmaster operators attempt to track which ISPs can control each node to gain an informal intuition of the independence of the network [2]. Previous anonymity networks, such as Tarzan and MorphMix, attempt to provide collusion resistance by comparing the IP of each peer [19, 33] (our results show that this technique is less effective than claimed). In this paper, we evaluate the topologies of *real anonymity networks* in the context of the properties of Internet routing at the AS-level, and design ways to quantify the results.

2.2 Internet Routing and Topology

To determine the networks that packets will traverse between each node of a mix network, we must first understand how packets

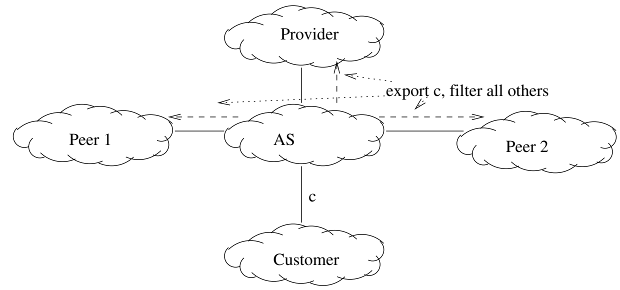

Figure 1: Common relationships and export restrictions.

are routed between two arbitrary hosts on the Internet. In this section, we first present a brief overview of interdomain routing (i.e., routing between ISPs) on the Internet and then describe available data on Internet topologies and our assumptions regarding how well this data reflects the paths that packets actually travel.

2.2.1 Border Gateway Protocol

The Internet is composed of about 17,000 independently operated networks, or autonomous systems (ASes), that exchange reachability information using the Border Gateway Protocol (BGP) [32]. An AS could be an Internet Service Provider (ISP), a corporate network, or a university. Each AS has a network of routers that route traffic to global destinations using the information propagated by routing protocols. To find the route to a destination IP address, a router performs a "longest prefix match" on that IP address to find the most specific IP prefix in the routing table that contains that IP address. For example, to look up *IP address* `18.31.0.82`, a router might use a route for the *prefix* `18.0.0.0/8`. The router then forwards packets for that destination to the next hop specified for the route to the prefix. Routers will select the route that is the *smallest* prefix that contains the IP address; for example, if a router's routing table had a prefix for, say, `18.31.0.0/16`, that router would prefer this route over the former.

The Internet's routing table has over 130,000 prefixes, each of which has an associated route. An AS that originates a route advertises this route to neighboring ASes via BGP and attaches its AS number to the *AS path* of the route. When a router in a neighboring AS learns this route, that router propagates it to all of the routers in the AS. Some of these routers will, in turn, exchange routes with other ASes. A router will typically readvertise the route to neighboring ASes, prepending its own AS number to the AS path in the process. In this fashion, BGP allows each AS to learn the AS-level path of a route to a destination that it learns via BGP.

ASes do not blindly propagate routes to all of their neighbors; rather, each pair of ASes has a commercial relationship, and an AS may prefer to send traffic via one AS over another for economic reasons. ASes form bilateral arrangements that can be broadly categorized as either (1) a *customer-provider* relationship, where the customer pays the provider to route traffic for it; or (2) a *peering* relationship, where two ASes exchange traffic between their own networks (and the networks of their customers), but neither pays the other for this privilege.

BGP routing is based on *policy*, not on shortest paths. For example, the AS in Figure 1 will typically prefer to route traffic to a destination via one of its customers (who pays it for connectivity) than via one of its providers (whom it must pay to send traffic toward) or one of its peers. These relationships also determine which routes one AS will advertise to another—an AS will not typically advertise a route learned from one of its peers or providers to any of

```
   Network      Next Hop        Metric LocPrf  Path
*>i18.0.0.0/8  64.243.30.141              100  6347 3356 3 i
* i            65.115.97.141       10     100  209 10578 3 i
```

Figure 2: Example BGP routing table entry (taken from a Cisco-like router).

its other peers or providers: doing so would constitute an implicit agreement to forward traffic between two of its providers, two of its peers, etc. The AS in Figure 1 would advertise routes learned from its customer to all of its neighbors, but would not readvertise routes learned from Peer 1 to Peer 2 (and vice versa), nor to its provider. It would also advertise the routes learned from its provider to its customer, but not to other peers.

Figure 2 shows a simplified BGP routing table entry. This router has learned two routes to the destination prefix `18.0.0.0/8`. Each route has various attributes, including the "next hop" IP address (where to route packets that use this path), various attributes that affect which route is selected as the preferred route to the destination, and the AS path ("Path"). The "$>$" at the beginning of the first line indicates that the router has selected this route as the best route to the destination using the BGP decision process.

Each router can only have a single best route to a destination at any time. This routing table entry allows us to be reasonably certain that a packet that is destined for the destination IP address `18.31.0.38` will traverse the networks corresponding to AS numbers 6347 (Savvis), 3356 (Level 3), and 3 (MIT). Packets tend to follow this sequence of ASes since, at the AS level, traffic flows in the opposite direction in which routers advertise the routes.[1]

2.2.2 *AS-level Internet Topology*

Paths between end-hosts in the Internet traverse a sequence of ASes (or jurisdictions); to estimate the sequence of ASes that any given path crosses, we must first have a representation of the Internet topology at the AS-level (i.e., the ASes that each AS connects to, as well as their business relationships). Determining a complete view of the AS-level graph is notoriously difficult, because bilateral policies hide edges in the graph from some perspectives [11]. For example, in Figure 1, a routing table captured at Peer 1 will not contain any routes with the $AS \leftrightarrow$ Peer 2 link, since the AS in the center will not readvertise routes learned from one peer to another.

There are many publicly available places that provide access to routing table data. The most prevalent is the Oregon RouteViews Project [28], which maintains a route server that peers with more than 50 ASes. Each of these ASes sends its routing tables to the RouteViews server, which learns that AS's best route to each destination prefix. Each AS's routing table is slightly different, which means that the AS-level topology constructed from the RouteViews route server is missing some inter-AS edges due to bilateral policies, but the graph is representative enough for our purposes. In the future, we could improve our analysis by incorporating newer techniques for capturing AS-level topologies [11]. Knowing the AS-level topology is not enough to determine the AS-level path between two arbitrary mix nodes, though; to determine this, we need to make some assumptions about the AS-level paths that packets actually traverse, which we describe in Section 4.2.

[1] There are some rare exceptions to this rule. For example, discrepancies can result if a router that advertises a BGP route via one AS "deflects" data packets to a router within that AS that has selected a different next-hop AS[21] (note that this is a routing protocol *misconfiguration*). Additionally, recent work has observed that the AS path in the routing table may not always match the sequence of networks that a packet is forwarded through, but typically the differences are minor and occur infrequently [25].

3. THREAT MODELS

Alice wants to communicate with Bob without revealing her location. We intend to improve Alice's anonymity against an adversary who can monitor a single AS (for example, a curious ISP or a corrupt law enforcement officer abusing his subpoena powers). We assume that the ability to observe multiple ASes is significantly more difficult than observing a single AS, because most ISPs do not control multiple ASes, and because law enforcement will be less willing to face the increased accountability and risk associated with obtaining multiple unapproved subpoenas.

To investigate further, we must consider which attacks are most effective against different classes of anonymity networks. We divide attacks into intra-network attacks and endpoint attacks, as described in Section 2.1.

Endpoint attacks on low-latency networks are the most straightforward: an adversary observing both Alice and Bob can quickly learn that they are communicating. Previous analysis of Onion Routing has shown that an adversary observing c of the n nodes in the network can break $(c/n)^2$ of the transactions [36]. By requiring the path from Alice to the anonymity network and the path from the anonymity network to Bob to traverse separate ASes, we can prevent all of these observed transactions as long as the ASes do not collude.

Intra-network attacks on low-latency networks can also be useful. In particular, paths in Tor and the (no longer deployed) Freedom protocol are generally 3 hops—short enough to maintain usability, but not so short that two nodes can be certain of linking Alice to Bob if they decide to collude [4, 18]. An adversary who can observe two links on the path breaks this assumption. If such an adversary is common, these designs should reconsider path length.

A successful endpoint attack against a high-latency system like Mixmaster takes a lot more time and effort than one against a low-latency system like Tor. However, because an observer of even a few Mixmaster nodes may be able to link Alice to her recipients over time [26], our work is also relevant for protecting such high-latency systems from a single-AS adversary. Further, intra-network observations (particularly during periods of low traffic) can be combined with active attacks such as message flooding to shrink the set of messages that mix with Alice's message [9, 17].

4. MODELING TECHNIQUES

We now describe how we model mix networks and Internet routing to draw conclusions about an anonymity network's vulnerability to eavesdropping by the adversary detailed in Section 3. First we describe our model of node selection in a mix network. Then, we present our techniques for estimating the AS-level path between two arbitrary hosts on the Internet.

4.1 Node Selection in Mix Networks

To establish a path in an anonymity network, clients must somehow discover a set of current nodes. In Mixmaster, clients examine the output of "pinger" software that measures node reliability and publishes keys and addresses for each node [29]. In Tor, clients download a similar network snapshot from special nodes called directory servers [18]. The pingers and directory servers note whether each node is an *exit node*—meaning its operator is willing to allow traffic to exit the network from the node (some operators choose instead to be *middleman* nodes, to avoid needing to deal with abuse complaints.)

We abstract how Alice gets the list: assume she has a set N of possible choices, of which $E \subseteq N$ are exit nodes. Also assume that all nodes in the network are listed as working (typically some nodes are listed as temporarily offline).

To build a path of length ℓ, Alice first selects an exit node at random from E, and then selects the other $\ell - 1$ nodes from N. In the *remailer network* case she selects nodes such that no node appears twice in a row; in the *onion routing* case she selects nodes such that no node appears twice anywhere in the path.

4.2 AS-level Mix Network Path Estimation

Active measurement tools such as "traceroute" could be used to discover AS-level paths. For example, the mix network operator could execute traceroutes between each pair of mix nodes to determine the IP-level paths (and, hence, the AS-level paths) between them. First, note that these measurements would not be robust against single compromised mixes. More importantly, however, Alice must *also* determine the AS-level path between herself and the mix entry she selects, as well as the AS-level path between the mix exit she selects and the destination where she is sending packets. To discover the AS-level path between herself and a good candidate mix node, Alice must run traceroutes to nodes in the mix network, which may engender suspicion. Further, she will not be able to actively determine the AS-level path from her chosen exit node and her destination: routing tables at that node may be unavailable or difficult to obtain covertly, and a traceroute from candidate exit node to the destination is also likely to engender suspicion (this approach will not work anyway if the node has been compromised). Finally, without access to a host at the destination node, Alice will be unable to run a traceroute from the destination node to her chosen exit node (i.e., the path that traffic from the destination to Alice will traverse): in this case, Alice can only discover the AS-level path from the destination to her chosen exit node using passive inference techniques, such as examining routing tables.

If Alice had access to an up-to-date routing table from every network containing mix nodes, she could construct a reasonable estimate of the AS-level path fairly easily: to discover the AS-level path between nodes i and $i + 1$, for example, she could look at i's routing table and determine the AS path associated with the route that is the longest prefix match for $i + 1$'s IP address.

Unfortunately, Alice cannot ask for routing tables for each of the mix nodes when constructing a mix tunnel. First, her act of requesting a routing table from a particular network might attract the attention of an eavesdropper, particularly if she asks for a large number of routing tables. Second, asking each network that contains a mix node for its current routing table is likely to be quite slow, since each full routing table is approximately 10 megabytes; additionally, as routes are continually changing, parts of the table are likely to be out-of-date even before she requests it. Third, this method introduces another vulnerability to attack: an adversary who compromises any of the domains that contain a mix node could send back an inaccurate version of the routing table.

Because of these shortcomings, Alice must *passively* determine the AS-level path (or a reasonable approximation of it) without having visibility into the routing tables of each hop in her intended mix path. Fortunately, examining the AS paths in a BGP routing table gives a reasonable estimation of what ASes connect to what other ASes, and can provide reasonable information about what path an arbitrary Internet host might take to reach any given destination.

We now summarize an AS-level path estimation technique that is based on the technique recently proposed by Mao *et al.*[24] Although it is admittedly impossible to determine an AS's routing policy with absolute certainty, Mao's work suggests that inferring AS-level paths based on common policies is accurate for more than 80% of paths.

1. *From one or more BGP routing tables, construct an AS-level graph representing the Internet's topology.* Routes in BGP routing tables have an AS path attribute, which provides a list of AS adjacencies. For example, from the routes in Figure 2, we know that AS 3356 and AS 3 are directly connected. Given the complete list of adjacencies from a BGP routing table, we can reasonably approximate the AS-level topology of the Internet.

   Of course, because the policies are applied based on commercial relationships (e.g., an AS may filter routes learned from one peer when advertising routes to another peer or provider), certain edges in this graph will not be globally visible. As a result, our approximation of the AS-level graph may omit certain edges. Typically, these missing edges will be between smaller ASes; thus, our algorithm may not realize that a particular edge exists between two ASes and, as a result, infer the wrong AS-level path to a destination.

2. *Determine the origin and destination ASes for the path in question.* To determine the AS-level path between two hosts, we must first determine the ASes where the hosts are located. This is reasonably easy: generally, it is sufficient to look in a BGP routing table and find the final AS in the AS path for a particular destination. For example, in Figure 2, the last AS in each AS path to the prefix `18.0.0.0/8` is AS 3 (MIT); therefore, it is generally safe to assume that any prefix contained within `18.0.0.0/8` is located in AS 3.

   Because ASes often allocate address space to their customers from their own address space, this technique should be applied to the most specific prefix in the routing table.

3. *Determine the relationships between each pair of ASes.* This is a notoriously difficult problem, because ASes typically guard the nature of the relationships they have with neighboring ASes. Fortunately, we can use heuristics from previous work that tend to work reasonably well [20].

   The basic idea is to exploit the *valley-free* property of Internet paths to assign pairwise relationships between ASes. That is, an AS path traverses a sequence of customer-provider edges, zero or one peering edges, and then a sequence of provider-customer edges. Therefore, each AS pair in an AS path can be assigned either a customer-provider or a provider-customer relationship: every pair before the AS with the highest degree in the path is assigned a customer-provider relationship, and every pair after this AS is assigned a provider-customer relationship. If, for two separate AS paths, two ASes are customers of each other, then the algorithm designates them as peers. The complete details of the inference algorithm are provided in previous work [20].

4. *Estimate the AS-level path between the two ASes by finding the shortest AS path that complies with common policy practices.*

   Because BGP routers select a single best route to each destination, *each pair of hosts will typically traverse a single, unique AS path in each direction*. (See Section 2.2.1 for a discussion of exceptions.) This step assumes that ASes implement policy that prefers the shortest AS path that is consistent with the best common practice of preferring customer routes over peering routes and peering routes over provider routes. Mao *et al.*'s algorithm suggests that this assumption is reasonable.

As AS-level path estimation techniques improve and techniques to estimate the actual AS-level forwarding path mature [25], determining ASes that mix networks traverse will become easier. These improvements will allow Alice to make informed decisions about the mix nodes she should choose to achieve location independence (it will also improve the accuracy of the type of analysis we present in this paper).

Given both a model for how anonymizing networks select nodes and an estimation of the AS-level path between two arbitrary hosts on the Internet, Alice can determine the complete set of ASes that a typical mix network path traverses using only passive techniques.[2] We explore these questions in further detail in Section 6.

5. DATA

In this section, we summarize the data that we use in our analysis of AS-level paths in mix networks. We base our analysis on the location of mix nodes in deployed systems today. We then describe the data we used to generate the AS-level network topology.

5.1 Mix Networks, Senders, and Receivers

To evaluate node selection in the Mixmaster and Tor models, we use operational mix nodes for each respective network; the tables in Appendix B provide lists of mix nodes for the two networks.

Since we are also interested in the AS-level paths between Alice and the mix entry point, and between the mix exit point and Bob, we must also estimate the ASes where Alice and Bob may typically be located. Unfortunately, usage data for these mix networks is not readily available, so it is not possible to drive our simulation with lists of common locations of senders and receivers. Nevertheless, we can perform reasonable approximations by assuming that Alice is located on a home network (e.g., a cable modem network, a DSL network, etc.) and that Bob is a content host located at a data hosting ISP.

To generate a set of ASes where senders might be located, we created a list of DSL and cable modem providers from `www.dslreports.com` that would be likely senders and mapped these providers to their respective AS numbers. To generate a list of typical receivers, we sampled sites from comScore Media Metrix's "Top 50 US Internet Properties" from December 2003 [13], as well as sites that we think might be popular on anonymity networks. The lists of senders and receivers that we used in our experiments are in Appendix A.

In this paper, we use the topologies of existing mix networks to get a plausible set of nodes for our model. The Tor nodes represent a newborn network where the only participants are developers and very early adopters, whereas the Mixmaster network represents a wider participant set because it has been deployed for many years. We compare how each of these node sets performs when the initiators are typical DSL or cable modem users in the US, and the responders are popular websites in the US—in effect, we are evaluating the location independence of the newborn Tor network and the independence of a node set that should resemble that of Tor network as Tor matures.

5.2 Internet Topology

To generate an estimate of the Internet's AS-level topology, we use the routing table dump from the `route-views.oregon-ix.net` route server on January 25, 2004 at 10:22 p.m. GMT. The table has 67 external BGP (eBGP) feeds from 53 ASes (some ASes have multiple eBGP feeds to the route server). We use this table to (1) generate our view of the AS-level topology, including inter-AS relationships, and compute pairwise AS-level shortest paths, as we described in Section 4.2 and (2) map IP addresses to the ASes where they are located.

[2]We performed our analysis in Section 6 using this passive technique because we could not run traceroutes between the Mixmaster nodes, and we wanted to directly compare the Tor and Mixmaster networks. As part of our future work, we plan to use `traceroute` to measure pairwise paths on the Tor network and compare the accuracy of the AS-level estimations that Alice would make using this technique against the "ground truth".

6. RESULTS

First, we discuss the fundamental robustness properties of existing mix networks. In addition to maximizing location independence at the entry and exit points of the mix network, we should try to minimize the cases where one AS can observe multiple links along a mix network path. This analysis is independent of our model for mix network users (i.e., senders and receivers), since we are only examining properties of the mix nodes themselves. Next, we compute the probability that the AS-level path from the sender to the entry node and the path from the exit node to the receiver traverse the same AS (i.e., the probability that a single AS can observe both endpoints of a mix network path), given the Tor and Mixmaster topologies and reasonable assumptions about the locations of senders and receivers.

6.1 Independence of Mix Nodes and Paths

In this section, we explore and quantify the location independence of the Mixmaster and Tor topologies. We examine cases where Tor and Mixmaster nodes are located in the same AS. We also examine the AS-level path properties between pairs of existing mix nodes and quantify the extent to which the AS-level paths between two mix nodes traverse common ASes. We examine the likelihood of mix-level paths traversing common ASes in both the forward (i.e., sender to recipient) and reverse (i.e., recipient's reply to sender) directions.

6.1.1 Node properties

The tables in Appendix B show that both the Mixmaster and Tor networks have multiple nodes in the same AS. Tor has three mix nodes in AS 23504 (Speakeasy DSL), and Mixmaster has two nodes each in ASes 3269 (Telecom Italia), 6939 (Hurricane Electric), 7132 (SBC), 23504 (Speakeasy DSL), and 24940 (Hetzner Online). This lack of location independence in node placement is not surprising; in particular, it reflects the fact that these network nodes are operated by *volunteers*, many of whom commonly operate mix nodes from their Internet connections at home (e.g., DSL providers, etc.). Nevertheless, the fact that both of these networks have multiple duplicate ASes suggests that users of these mix networks should exercise caution when selecting mix nodes (particularly the entry and exit nodes).

Previous work (and conventional wisdom) has suggested that selecting nodes from disjoint subsets of the IP address space will achieve independence in node placement; it is clear from our survey of Mixmaster and Tor that these types of prefix-based mechanisms are, in general, ineffective, and they can give the user a false sense of security. For example, Tarzan and MorphMix suggest subdividing the node space into `/16` prefixes, and subsequently into `/24` prefixes and selecting nodes from distinct subsets of the IP prefix space to reduce the likelihood that two mix nodes are in the jurisdiction of a single AS [19, 33]. Unfortunately, this technique does not necessarily increase the likelihood of location independence: of the five pairs of Mixmaster nodes that are located in the same AS, three of these pairs (those in ASes 3269, 7132, and 23504) not only have distinct `/16` prefixes, they also have distinct `/8` prefixes. Similarly, one of the Tor network nodes in AS 23504 has a distinct

	Tor		Mixmaster
# of AS-disjoint mix node pairs	961		1764
# of mix node pairs with common AS			
AS 3356 (Level 3 Communications, LLC)	276 (28.7%)	AS 3356 (Level 3 Communications, LLC)	291 (16.5%)
AS 6461 (Abovenet Communications, Inc)	249 (25.9%)	AS 6461 (Abovenet Communications, Inc)	251 (14.2%)
AS 2914 (Verio, Inc)	65 (6.8%)	AS 7018 (AT&T WorldNet Services)	234 (13.3%)
AS 16631 (Cogent Communications)	64 (6.7%)	AS 3549 (Global Crossing)	104 (5.9%)
AS 701 (UUNET Technologies, Inc)	61 (6.3%)	AS 14188 (Ashland Fiber Network)	82 (4.6%)
AS 23342 (United Layer, Inc)	60 (6.2%)	AS 23342 (United Layer, Inc)	82 (4.6%)
AS 19782 (Indiana University)	60 (6.2%)	AS 1668 (AOL Transit Data Network)	82 (4.6%)
AS 2152 (California State University)	60 (6.2%)	AS 15290 (Allstream Corp. Corporation Allstream)	49 (2.8%)
AS 10578 (Harvard University)	53 (5.5%)	AS 2914 (Verio, Inc)	46 (2.6%)
AS 3491 (CAIS Internet)	52 (5.4%)	AS 6993 (Internap Network Services)	42 (2.4%)

Table 1: Characterizing location independence in Mixmaster and Tor.

/8 prefix. To achieve location independence, a mix network must explicitly consider the actual AS of a host, not simply its IP address.

Finally, we note that many of the Tor network's exit nodes are currently located in the United States. In practice, this network could achieve greater location independence by increasing exit node participation outside of the US.

6.1.2 Path properties

Table 1 shows the extent of location independence in Mixmaster and Tor. Tor has 35 nodes that are located in 31 distinct ASes, for a total of 961 AS-disjoint mix node pairs; similarly, Mixmaster has 49 nodes located in 42 distinct ASes, or 1764 AS-disjoint node pairs. The most striking statistic is that AS 3356 appears on 276, (nearly 30%) of Tor's AS-disjoint paths; AS 3356 also appears on about 17% of Mixmaster's AS-disjoint paths between node pairs. The reason for this prevalence can be explained by two factors: (1) the location of nodes in the mix network, and (2) fundamental properties of the AS-level topology (i.e., many paths ultimately traverse some tier-1 ISP; those in the mix topologies we examined seem particularly likely to traverse AS 3356).

First, many of both Tor's and Mixmaster's nodes are located in *edge* networks; this means that, for some nodes, the path both to and from that node will cross the same AS much of the time. This phenomenon is especially true for nodes that are located on edge networks with a single preferred upstream ISP; for example, the nodes at MIT use AS 3356 for most inbound and outbound paths, with the exception of paths to and from Internet2 destinations.

Second, many paths in the Internet, particularly those between two edge networks, will traverse at least one large "tier-1" ISP (i.e., an ISP that operates its own backbone and does not buy upstream service from another ISP). Not surprisingly, Table 1 shows that many of the ASes that are between a large number of mix node pairs are tier-1 ISPs (e.g., UUNet, Qwest, Global Crossing, AT&T, AOL, Verio, and Abovenet).

The prevalence of certain ISPs between mix node pairs suggests that as the length of a mix network path increases, the likelihood that an AS will be able to observe the path at more than one location also increases. Still, the likelihood that an AS should be able to observe a significant fraction of the links on a mix network path should decrease as the length of the path increases. To test this hypothesis, we generated random mix paths through the mix network. Using both the *remailer* and *onion routing* node selection algorithms (as described in Section 4.1), and varying lengths from two hops to eight hops, we measured the probability that a path crosses the same AS on multiple links. For each length and type of path, we ran 10,000 trials.

Figure 3 shows the probability that a single AS will be able to observe all of the links along the mix network path, for mix network paths of different lengths. Figure 4 shows the probability that a single AS will be able to observe all but one of the links along a path of a certain length. (Figures 5 and 6 show the same properties for the *reverse* paths through the mix network.) Paths of length one and two have less than two links and, thus, are never observed by the same AS twice. The AS that contains the second node in a three-hop path will always observe all links in the path because it is incident on both links in the path; for the same reason, the ASes of the second and third hops in a four-hop path will always be able to observe all but one link in the path.

The figures show results for both the Tor and Mixmaster network topologies, with two different node selection schemes: (1) allowing the same mix node to be used twice along the mix path, as long as the same mix node is not used for two consecutive hops ("with replacement", as in *remailer networks*) and (2) allowing each mix node to be used only once ("without replacement", as in *onion routing*). Figure 3 shows two interesting results. First, for all mix paths shorter than four hops, a single AS can observe all of the links in the mix network path. Second, Tor's node selection algorithm (i.e., the onion routing scheme) provides significant protection against observation at multiple links for both the Tor and Mixmaster network topologies. For example, a four-hop path constructed from Tor nodes without node replacement will be observed by a single AS on all links with probability 0.10, whereas a four-hop path constructed with node replacement will be observed with probability 0.16. This result makes sense: random node selection with replacement is more likely to result in the same hop being used twice along a single mix path, if this is not explicitly prevented. Figures 5 and 6 also seem to indicate that reverse paths through the mix network (i.e., paths from Web servers to cable modem-type users) are slightly more vulnerable to observation on both entry and exit than vice versa.

6.2 Independence of Entry and Exit Paths

To evaluate the location independence of the entry and exit paths for typical mix networks, we used the lists of common sender and receiver locations from Appendix A and modeled typical paths from the sender to receiver through both the Mixmaster and Tor topologies.

To do this, we generated 10,000 random entry and exit pairs for each network and, for each sender/receiver pair, observed the number of times the path from the sender to the entry node traversed at least one AS on both paths; we performed this analysis for both forward and reverse paths through the mix network. Tables 2 and 3 show the probability, for each sender and receiver, of this event. We see that each pair of sender and receiver has at least some subset of entry and exit paths that traverse the same AS upon both entry and

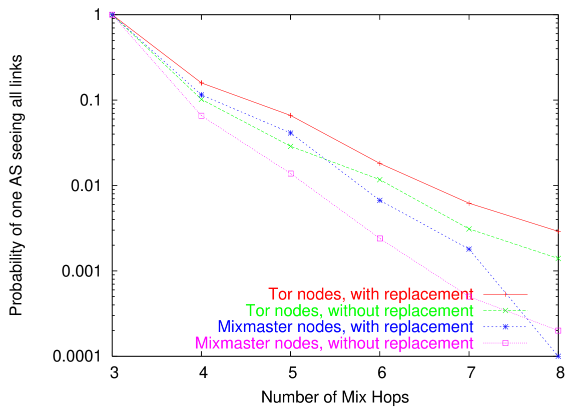

Figure 3: Fraction of paths where a single AS can observe all links in the mix network path.

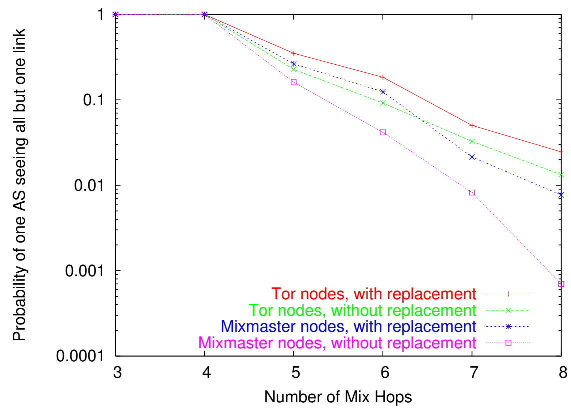

Figure 4: Fraction of paths where a single AS can observe all but one links in the mix network path.

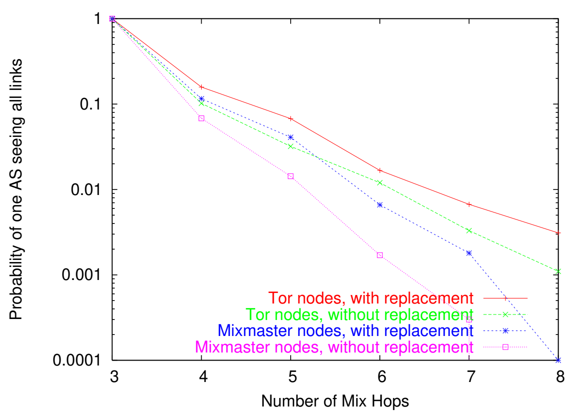

Figure 5: Fraction of paths where a single AS can observe all links in the *reverse* mix network path.

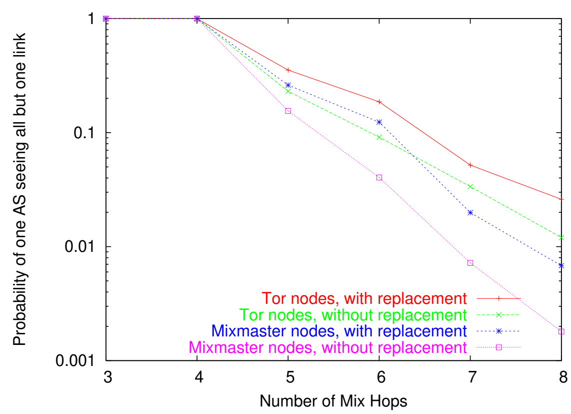

Figure 6: Fraction of paths where a single AS can observe all but one links in the *reverse* mix network path. (*Note:* slightly different y-axis scale.)

exit. Upon further investigation, we learned that the AS that was traversed on both entry and exit most often was *always* a tier-1 ISP.

These results suggest that the sender in a mix network should exercise care when selecting entry and exit nodes to avoid choosing entry and exit paths that traverse the same AS. They also suggest that it is certainly *possible* for an intelligent sender to select entry and exit nodes such that the entry and exit paths do not traverse the same AS on entry and exit (e.g., between Speakeasy and Google, only 7% of Tor entry/exit node pairs result in entry and exit paths that cross the same AS on both entry and exit). However, a careless sender that does not pay attention to the AS-level topology may well be observed by a single AS at both entry and exit. For example, if Alice uses AOL (AS 1668) as her ISP and attempts to connect to `cnn.com` (AS 5662), a single AS (i.e., AS 1668) will observe both the entry and exit paths with absolute certainty, because AOL Time Warner owns Turner Broadcasting (AS 5662), which includes CNN.

Location independence for pairs of senders and receivers can be highly asymmetric. For example, in the Tor network, from Comcast (AS 22909) to indymedia (AS 22489), 45% of the entry/exit node pairs result in paths that traverse the same AS on both entry and exit; from indymedia to Comcast, on the other hand, random entry and exit node selection is susceptible to observation on both paths in only 9% of cases. This result suggests that, in certain cases, *a user may wish to establish different mix-level paths for forward and reverse traffic* to minimize the possibility that a single AS can observe both entry and exit traffic. This finding is not entirely unexpected, given the asymmetric path properties of the Internet.

Interestingly, these tables also show that location independence is high when the sender, the receiver, or both are located in a tier-1 ISP (e.g., AS 4999, which is part of Sprint). This might be because the path from the sender to the entry point is already located in a tier-1 ISP, and thus will not have to cross other tier-1 ISPs en route to the entry point.

7. DESIGN RECOMMENDATIONS

In light of our analysis, which has shown that certain ASes have considerable eavesdropping capabilities on mix networks, we propose two recommendations with regard to mix network design. First, mix networks should select paths with the underlying AS-level topology in mind. Second, mix networks should strive to deploy more nodes in locations with rich connectivity to other ASes.

7.1 Consideration of AS-level Paths

Our results suggest that designers and users of mix networks should take into account the underlying AS-level paths of each link in the mix network. Mix network paths can be made more safe if

	Receiver													
Sender	2914	4323	5662	7224	7784	10593	11643	12076	12182	15130	15169	17110	22489	26101
209	0.17 **(0.09)**	**0.07** (0.13)	0.13 (0.14)	**0.08** (0.14)	**0.05** **(0.09)**	0.13 (0.14)	0.18 (0.22)	**0.08** (0.14)	**0.09** (0.13)	0.15 **(0.09)**	**0.06** (0.13)	0.17 **(0.09)**	0.13 (0.11)	0.15 (0.14)
1668	0.16 **(0.09)**	**0.08** (0.11)	*1.00* *(1.00)*	**0.08** **(0.07)**	0.10 **(0.08)**	*1.00* *(1.00)*	0.15 **(0.09)**	0.19 **(0.09)**	0.10 (0.15)	0.13 **(0.04)**	**0.09** **(0.09)**	0.27 **(0.08)**	0.14 (0.12)	0.25 (0.18)
4355	**0.08** (0.10)	**0.05** (0.14)	**0.04** **(0.09)**	**0.01** (0.20)	**0.06** **(0.06)**	**0.04** **(0.09)**	**0.08** **(0.09)**	0.12 (0.11)	**0.06** **(0.08)**	**0.03** **(0.03)**	**0.08** (0.13)	0.17 (0.10)	**0.08** **(0.06)**	0.16 (0.18)
4999	0.11 **(0.06)**	**0.03** **(0.08)**	**0.04** (0.13)	0.42 (0.26)	**0.04** **(0.05)**	**0.04** (0.13)	0.20 (0.10)	0.32 (0.13)	0.11 **(0.06)**	0.34 *(0.86)*	**0.03** **(0.07)**	0.11 **(0.05)**	0.25 (0.20)	0.42 (0.25)
6079	0.16 (0.13)	**0.09** (0.14)	0.10 (0.11)	**0.03** **(0.08)**	0.18 (0.40)	0.10 (0.11)	0.17 (0.12)	0.22 (0.13)	0.11 (0.10)	**0.05** **(0.03)**	0.14 (0.18)	0.33 (0.16)	0.14 **(0.07)**	0.29 (0.34)
6995	0.19 (0.12)	0.11 (0.10)	0.14 (0.18)	**0.09** **(0.08)**	0.12 **(0.09)**	0.14 (0.18)	0.19 (0.14)	0.18 (0.22)	0.14 (0.16)	0.15 **(0.06)**	0.12 **(0.09)**	0.28 (0.14)	0.17 (0.12)	0.25 (0.38)
18566	0.27 (0.27)	0.22 (0.26)	0.23 (0.38)	**0.08** (0.17)	0.26 (0.24)	0.23 (0.38)	0.36 (0.29)	0.50 (0.35)	0.24 (0.38)	0.18 (0.13)	0.29 (0.19)	0.74 (0.31)	0.34 (0.29)	0.67 *(0.86)*
22773	0.13 (0.10)	0.13 (0.16)	0.11 **(0.09)**	**0.03** (0.21)	0.13 **(0.06)**	0.11 **(0.09)**	0.19 **(0.09)**	0.25 (0.11)	0.11 **(0.08)**	**0.06** **(0.03)**	0.17 (0.15)	0.36 (0.10)	0.16 **(0.06)**	0.33 (0.18)
22909	0.14 **(0.07)**	0.21 (0.11)	0.12 (0.14)	0.49 (0.66)	0.31 **(0.06)**	0.12 (0.14)	0.10 **(0.07)**	0.13 **(0.07)**	0.29 (0.10)	0.17 **(0.03)**	0.13 (0.11)	0.17 **(0.07)**	0.45 **(0.09)**	0.16 (0.15)
23504	0.15 (0.15)	**0.06** **(0.04)**	**0.07** (0.13)	**0.07** (0.22)	**0.06** **(0.07)**	**0.07** (0.13)	0.10 (0.11)	0.14 (0.11)	**0.08** (0.14)	0.31 **(0.09)**	**0.07** **(0.04)**	0.18 (0.12)	0.11 (0.12)	0.19 (0.23)

Table 2: Location independence for typical sending and receiving ASes for forward (and reverse) paths in the Tor network topology. Each entry shows, for a sender/receiver pair, the probability that a single AS will observe both the path from the sender to the entry node and the path from the exit node to the receiver. Names for each AS are listed in Appendix A.

	Receiver													
Sender	2914	4323	5662	7224	7784	10593	11643	12076	12182	15130	15169	17110	22489	26101
209	**0.07** **(0.07)**	**0.06** **(0.08)**	**0.09** **(0.08)**	**0.09** (0.10)	0.11 **(0.06)**	**0.09** **(0.08)**	0.16 (0.12)	0.14 **(0.05)**	**0.09** **(0.06)**	**0.08** (0.17)	**0.07** **(0.07)**	0.18 **(0.07)**	0.11 **(0.08)**	0.17 (0.13)
1668	0.10 **(0.08)**	**0.06** (0.10)	*1.00* *(1.00)*	0.11 (0.11)	0.10 **(0.07)**	*1.00* *(1.00)*	0.15 **(0.08)**	0.17 **(0.07)**	0.11 **(0.07)**	0.11 (0.14)	**0.06** **(0.08)**	0.19 **(0.07)**	0.14 **(0.09)**	0.20 (0.16)
4355	**0.07** **(0.07)**	**0.05** (0.10)	**0.08** (0.12)	**0.08** (0.26)	**0.07** **(0.06)**	**0.08** (0.12)	**0.09** **(0.07)**	0.10 **(0.07)**	**0.07** **(0.05)**	**0.09** (0.10)	**0.06** **(0.07)**	0.10 **(0.07)**	**0.09** **(0.07)**	0.12 (0.13)
4999	0.18 (0.10)	0.16 (0.14)	0.18 (0.27)	0.40 (0.21)	0.10 **(0.09)**	0.18 (0.27)	0.28 (0.13)	0.32 (0.13)	0.23 **(0.09)**	0.47 *(0.81)*	0.11 (0.10)	0.14 (0.12)	0.32 (0.20)	0.40 (0.28)
6079	0.11 (0.12)	**0.07** **(0.09)**	0.10 **(0.08)**	**0.07** **(0.06)**	0.13 (0.36)	0.10 **(0.08)**	0.22 (0.11)	0.28 **(0.07)**	0.12 **(0.05)**	**0.08** (0.12)	0.10 **(0.07)**	0.31 **(0.09)**	0.15 **(0.06)**	0.32 (0.14)
6995	**0.07** **(0.08)**	**0.06** **(0.08)**	0.12 (0.12)	**0.06** **(0.06)**	0.10 **(0.05)**	0.12 (0.12)	0.20 (0.11)	0.23 (0.12)	0.11 **(0.08)**	**0.09** (0.11)	**0.08** **(0.09)**	0.27 (0.11)	0.14 (0.10)	0.27 (0.38)
18566	0.10 (0.13)	0.10 (0.14)	0.15 (0.22)	**0.06** **(0.09)**	0.16 **(0.09)**	0.15 (0.22)	0.43 (0.17)	0.58 (0.20)	0.21 (0.15)	0.11 (0.16)	0.18 **(0.07)**	0.64 (0.17)	0.27 (0.21)	0.67 *(0.84)*
22773	**0.09** (0.10)	**0.07** (0.11)	0.13 (0.10)	**0.06** (0.18)	0.10 **(0.04)**	0.13 (0.10)	0.24 **(0.06)**	0.32 **(0.07)**	0.13 **(0.06)**	0.10 **(0.09)**	0.12 **(0.07)**	0.33 **(0.07)**	0.17 **(0.08)**	0.37 (0.15)
22909	0.17 (0.11)	0.18 (0.12)	0.18 (0.21)	0.45 (0.70)	0.37 (0.13)	0.18 (0.21)	**0.08** (0.11)	0.10 (0.10)	0.22 **(0.09)**	0.14 (0.17)	**0.08** (0.11)	0.10 (0.12)	0.36 (0.12)	0.11 (0.15)
23504	**0.08** (0.12)	**0.05** **(0.06)**	**0.06** (0.11)	0.10 (0.15)	**0.05** **(0.08)**	**0.06** (0.11)	0.11 (0.12)	0.11 (0.10)	0.11 (0.12)	0.29 (0.24)	**0.04** **(0.05)**	0.12 (0.12)	0.12 (0.14)	0.14 (0.21)

Table 3: Location independence for typical sending and receiving ASes for forward paths through the Mixmaster anonymity network topology. Numbers in parentheses show location independence properties for *reverse* paths (i.e., traffic from receiver to sender).

senders increase the location independence of the paths they use, by explicitly choosing entry and exit nodes to avoid traversing the same AS upon entry and exit to the mix network.

However, while this approach is clearly better against a small adversary who owns one AS, we must also consider the effect against a large adversary who owns many ASes. By narrowing the set of possible mixes Alice might use, she gives *more* information to a large adversary. For example, an adversary who observes a transaction exiting the mix network at a Sprint node can deduce that Alice did not enter the mix network through a Sprint node. We must consider the effects of our suggested algorithm on all levels of adversary; we leave this investigation to future work.

7.2 Improved Node Placement

As mix networks expand, would nodes in certain ASes help to achieve diversity better than others? Our results suggest that nodes in edge networks (e.g., cable modem and DSL providers, universities, etc.) are likely to traverse the same AS on both the inbound and outbound paths to those nodes. Far-flung node locations that provide geographical diversity, such as nodes in Asia, are likely to actually *reduce* location independence, because such nodes do not typically have diverse AS-level connectivity. Rather, the best place for new nodes is likely to be in ASes that have *high degree*—that is, those that connect to a large number of other ASes. Ironically, the ASes with the highest degree tend to be tier-1 ISPs themselves; thus placing one node in each tier-1 ISP and building mix paths between those nodes may be the best strategy for increasing location independence. Exploring this question is an excellent direction for future work.

7.3 Other issues

Several other factors complicate our analysis, which we leave for future work. First, companies like Akamai provide Web hosting around the globe to serve content from locations that are close to any given user. They therefore present a challenge for this analysis. Because the exit node will choose a nearby Akamai server, Alice can no longer use the scheme in Section 4.2 to estimate the AS-level path between the exit node and her destination. Also, Akamai itself becomes a powerful global adversary with respect to certain popular websites. Second, more research remains to determine the sensitivity of our independence metric to the addition or removal of a few nodes in the topology. Third, our choice of popular locations for initiator and responder were all inside the United States. We should determine whether our analysis changes for users in foreign countries. Finally, for Alice to use this approach, she must periodically fetch routing tables and estimate the Internet's topology—which requires lots of computation and bandwidth. We must devise a way to condense this information; directory servers could then provide periodic signed snapshots.

8. CONCLUSION

We propose that mix networks should consider the underlying AS-level paths to achieve better location independence. Our paper presents several interesting and important results:

- While previous systems have proposed selecting nodes from disjoint IP address prefixes to select nodes in different jurisdictions, this technique is not sufficient to achieve location independence.
- Certain tier-1 ISPs are prevalent on many mix network paths. If node replacement is used in path selection, the probability that a single AS observes all links in a four-hop path through the mix is between 0.1 and 0.2; if node replacement is not used, this probability is no more than 0.1 for both the Tor and Mixmaster topologies.
- Given random entry and exit node selection, even when the initiator chooses distinct entry and exit nodes, a single AS will often be able to observe both the entry and exit path to the mix network between 10% and 30% of the time. *Because of path asymmetry in the Internet, an entry/exit node pair that has good location independence for a forward path through the mix network may not always have good location independence in the reverse direction.* However, if the initiator chooses entry and exit nodes with location independence in mind, she can prevent most such attacks.
- Figures 3 and 4 show that the intra-network diversity for the Tor topology is nearly equivalent to that of the Mixmaster topology. At least against observation attacks from a single AS, a newborn network with nodes almost entirely in the US is as robust as a mature network like Mixmaster.

9. REFERENCES

[1] Alessandro Acquisti, Roger Dingledine, and Paul Syverson. On the economics of anonymity. In Rebecca N. Wright, editor, *Financial Cryptography*. Springer-Verlag, LNCS 2742, 2003.

[2] Riot Admin. The remailer geographical mapping project. `http://riot.eu.org/anon/remap.html`.

[3] The Anonymizer. `http://anonymizer.com/`.

[4] Adam Back, Ian Goldberg, and Adam Shostack. Freedom systems 2.1 security issues and analysis. White paper, Zero Knowledge Systems, Inc., 2001.

[5] Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, *Information Hiding (IH 2001)*, pages 245–257. LNCS 2137, 2001.

[6] Krista Bennett and Christian Grothoff. GAP – practical anonymous networking. In Roger Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*. LNCS 2760, March 2003.

[7] Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Workshop on Design Issue in Anonymity and Unobservability*, pages 115–129. LNCS 2009, 2000.

[8] Oliver Berthold and Heinrich Langos. Dummy traffic against long term intersection attacks. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies (PET 2002)*. Springer-Verlag, LNCS 2482, 2002.

[9] Oliver Berthold, Andreas Pfitzmann, and Ronny Standtke. The disadvantages of free MIX routes and how to overcome them. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Workshop on Design Issue in Anonymity and Unobservability*, pages 30–45. Springer-Verlag, LNCS 2009, 2000.

[10] Philippe Boucher, Adam Shostack, and Ian Goldberg. Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc., December 2000.

[11] H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Towards capturing representative AS-level internet topologies. *Computer Networks Journal*, 2004.

[12] David Chaum. Untraceable electronic mail, return addresses, and digital pseudo-nyms. *Communications of the ACM*, 4(2), February 1981.

[13] comScore Media Metrix Announces Top 50 U.S. Internet Property Rankings for December 2003. `http://www.comscore.com/press/release.asp?press=402`, January 14, 2004.

[14] Wei Dai. Pipenet 1.1. Usenet post, August 1996. `http://www.eskimo.com/~weidai/pipenet.txt`.

[15] George Danezis. Statistical disclosure attacks: Traffic confirmation in open environments. In Gritzalis, Vimercati, Samarati, and Katsikas, editors, *Proceedings of Security and Privacy in the Age of Uncertainty, (SEC2003)*, pages 421–426, Athens, May 2003. IFIP TC11, Kluwer.

[16] George Danezis. The traffic analysis of continuous-time mixes. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies (PET 2004)*, May 2004.

[17] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *2003 IEEE Symposium on Security and Privacy*, pages 2–15. IEEE CS, May 2003.

[18] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[19] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, Nov 2002.

[20] Lixin Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, December 2001.

[21] Timothy Griffin and Gordon Wilfong. On the correctness of IBGP configuration. In *Proc. ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[22] Ceki Gülcü and Gene Tsudik. Mixing E-mail with Babel. In *Network and Distributed Security Symposium (NDSS 96)*, pages 2–16. IEEE, Feb 1996.

[23] Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew Wright. Timing attacks in low-latency mix-based systems. In Ari Juels, editor, *Financial Cryptography*. Springer-Verlag, LNCS 3110, 2004.

[24] Zhuoqing Morley Mao, Lili Qiu, Jia Wang, and Yin Zhang. Inferring AS-level paths with RouteScope. Technical Report TD-5T3RRP, AT&T Labs – Research, November 2003.

[25] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy Katz. Towards an accurate AS-level traceroute tool. In *Proc. ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[26] Nick Mathewson and Roger Dingledine. Practical traffic analysis: Extending and resisting statistical disclosure. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies (PET 2004)*, May 2004.

[27] Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster Protocol — Version 2. Draft, July 2003. `http://www.abditum.com/mixmaster-spec.txt`.

[28] University of Oregon. RouteViews. `http://www.routeviews.org/`.

[29] Peter Palfrader. Echolot: a pinger for anonymous remailers. `http://www.palfrader.org/echolot/`.

[30] J. F. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Workshop on Design Issue in Anonymity and Unobservability*. LNCS 2009, July 2000.

[31] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, May 1998.

[32] Y. Rekhter and T. Li. *A Border Gateway Protocol 4 (BGP-4)*. IETF, 1995. RFC 1771.

[33] Marc Rennhard and Bernhard Plattner. Practical anonymity for the masses with morphmix. In Ari Juels, editor, *Financial Cryptography*. Springer-Verlag, LNCS 3110, 2004.

[34] Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a trickle to a flood: Active attacks on several mix types. In Fabien Petitcolas, editor, *Information Hiding (IH 2002)*. LNCS 2578, 2002.

[35] Andrei Serjantov and Peter Sewell. Passive attack analysis for connection-based anonymity systems. In *Computer Security – ESORICS 2003*. Springer-Verlag, LNCS 2808, October 2003.

[36] Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an Analysis of Onion Routing Security. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Workshop on Design Issue in Anonymity and Unobservability*, pages 96–114. Springer-Verlag, LNCS 2009, July 2000.

APPENDIX

A. SUMMARY OF ENDPOINTS

Receivers and Senders used in Analysis

Receivers		Senders	
Receiver	AS	Sender	AS
www.cryptome.org	2914	Qwest	209
www.norml.org	2914	AOL	1668
www.anonymizer.com	4323	Earthlink	4355
www.cnn.com	5662	Sprint	4999
www.amazon.com	7224	RCN	6079
www.aclu.org	7784	Verizon	6995
www.aol.com	10593	BellSouth	12272
www.ebay.com	11643	Covad	18566
www.hotmail.com	12076	Cox	22773
www.hotornot.com	12182	Comcast	22909
www.dea.gov	15130	Speakeasy	23504
www.google.com	15169		
www.yahoo.com	17110		
www.indymedia.org	22489		
www.geocities.com	26101		

B. SUMMARY OF MIX NETWORKS

Mixmaster nodes as of June 2004
(exit nodes in boldface)

Name	IP address	Country	Autonomous System
lcs	18.26.0.254	US	3 (Massachusetts Institute of Technology)
willers	128.107.241.167	US	109 (Cisco Systems, Inc)
cf	**208.210.149.14**	**US**	**701 (UUNET Technologies, Inc)**
freedom	**205.241.45.100**	**US**	**1239 (Sprint)**
austria	**212.124.142.99**	**Austria**	**1901 (EUnet Austria)**
dizum	**194.109.206.210**	**Netherland**	**3265 (XS4ALL)**
george	**212.171.49.198**	**Italy**	**3269 (TELECOM ITALIA)**
starwars	**62.211.216.127**	**Italy**	**3269 (TELECOM ITALIA)**
nikto	**62.155.144.81**	**Germany**	**3320 (Deutsche Telekom AG)**
hastio	**80.34.205.8**	**Spain**	**3352 (Internet Access Network of TDE)**
cmeclax	208.150.110.21	US	3561 (Cable & Wireless USA)
itys	**209.221.142.117**	**US**	**3742 (Semaphore Corporation)**
cracker	207.15.209.4	US	4513 (Globix Corporation)
cripto	**195.250.236.58**	**Italy**	**5481 (ISET Informatica)**
bikikii	216.80.122.14	US	6079 (RCN Corporation)
bigapple	**167.206.5.3**	**US**	**6128 (Cablevision Systems Corp)**
aarg	69.9.134.82	US	6296 (InfoStructure)
banana	82.133.6.115	England	6728 (NILDRAM UK Peering)
randseed	**216.218.240.190**	**US**	**6939 (Hurricane Electric)**
liberty	216.218.240.134	US	6939 (Hurricane Electric)
anon	24.147.172.248	US	7015 (Comcast Cable Communications Holdings, Inc)
citrus	168.150.177.152	US	7132 (SBC Internet Services - Southwest)
cthulu	67.121.201.38	US	7132 (SBC Internet Services - Southwest)
congo	**216.154.65.55**	**Canada**	**7271 (Look Communications Inc)**
ashcroft	66.79.46.86	US	7776 (Commnet Data Systems, LLC)
hermes	208.42.19.154	US	8015 (Vector Internet Services, Inc)
rot26	62.245.184.24	Germany	8767 (M”net AS)
antani	**195.110.124.18**	**Italy**	**12363 (DADA S.p.a)**
amigo	212.67.202.215	England	12616 (Webfusion Internet Solutions Ltd)
riot	213.254.16.33	Italy	12779 (ITGATE.Net)
edo	**213.254.4.10**	**Italy**	**12779 (ITGATE.Net)**
paranoia	**213.140.29.37**	**Italy**	**12874 (Fastweb Autonomous System)**
panta	**217.155.84.182**	**England**	**13037 (Zen Internet)**
bunker	213.129.65.104	US	13108 (A.L. Digital Ltd. Kent site)
frell	**62.109.75.33**	**Germany**	**13184 (HanseNet Telekommunikation GmbH)**
lemuria	**213.191.86.35**	**Germany**	**13184 (HanseNet Telekommunikation GmbH)**
dot	81.0.225.26	Poland	15685 (Casablanca INT Autonomous system)
vger	66.166.203.164	US	18566 (Covad Communications)
dingo	208.180.124.28	US	19108 (Cox Internet Services)
chicago	65.31.179.120	US	20231 (HoldCo LLC - Road Runner)
tonga	**213.130.163.34**	**Netherland**	**20481 (Calyx Internet B.V. Netherlands)**
italy	**62.211.72.26**	**Italy**	**20580 (Telecom Italia Network)**
futurew	**212.66.104.81**	**Italy**	**20912 (Panservice)**
krotus	69.17.45.166	US	23504 (Speakeasy Inc)
harmless	66.92.53.74	US	23504 (Speakeasy Inc)
metacolo	**193.111.87.9**	**US**	**24812 (MetaColo AS)**
gbnq	213.133.98.183	Germany	24940 (Hetzner Online AG RZ-Nuernberg)
mercler	213.133.111.165	Germany	24940 (Hetzner Online AG RZ-Nuernberg)
discord	**141.12.220.23**	**Germany**	**28714 (Fraunhofer Gesellschaft (FHG)**

Tor nodes as of June 2004

(exit nodes in boldface)

Name	IP address	Country	Autonomous System
moria	**18.244.0.188**	**US**	**3 (Massachusetts Institute of Technology)**
cassandra	**140.247.60.133**	**US**	**11 (Harvard University)**
ovmj	**128.10.19.51**	**US**	**17 (Purdue University)**
nikitab	**128.32.37.191**	**US**	**25 (University of California at Berkeley)**
triphop	**152.2.241.23**	**US**	**81 (MCNC Center of Communications)**
randomtrash	**66.77.12.56**	**US**	**209 (Qwest)**
pvt	**128.100.171.30**	**CA**	**549 (ONet Networking)**
jap	141.76.46.90	DE	680 (DFN-IP service G-WiN)
hopey	**128.119.245.100**	**US**	**1249 (Five Colleges Network)**
code13	205.158.23.142	US	2828 (XO Communications)
peertech	**207.36.86.132**	**US**	**3064 (CyberGate Internet Technologies, Inc)**
dizum	**194.109.206.210**	**NL**	**3265 (XS4ALL)**
ubik	**194.109.217.74**	**NL**	**3265 (XS4ALL)**
itys	**209.221.142.117**	**US**	**3742 (Semaphore Corporation)**
tor26	62.116.124.106	AT	5424 (ATnet)
rootdown	**166.70.93.2**	**US**	**6315 (XMission)**
c3po	**128.187.170.212**	**US**	**6510 (Brigham Young University)**
bollox	**194.70.3.60**	**UK**	**6838 (Flirble IX)**
wannabe	**217.160.110.113**	**DE**	**8560 (Schlund + Partner AG)**
poblano	**129.170.19.228**	**US**	**10755 (Dartmouth College)**
mantaray	**209.142.37.21**	**US**	**10790 (InReach Internet)**
darkridge	**64.90.164.74**	**US**	**11403 (The New York Internet Company)**
rot52	**216.32.201.35**	**US**	**20473 (NetTransactions, LLC)**
Tonga	**213.130.163.34**	**NL**	**20481 (Calyx Internet B.V. Netherlands)**
anize	**69.56.216.138**	**US**	**21844 (THE PLANET)**
tequila	**216.27.178.156**	**US**	**23504 (Speakeasy Inc)**
nymip	**66.92.0.206**	**US**	**23504 (Speakeasy Inc)**
peerfear	**66.93.132.237**	**US**	**23504 (Speakeasy Inc)**
metacolo	**193.111.87.20**	**US**	**24812 (MetaColo AS)**
ned	**80.190.251.24**	**DE**	**24900 (IPX Server)**
petra	**69.20.9.201**	**US**	**27357 (Rackspace.com)**
TheoryOrg	**64.147.163.247**	**US**	**29752 (SFcolocation)**
incognito	**199.173.10.10**	**US**	**29944 (PullThePlug Technologies LLC)**