AS-awareness in Tor Path Selection

Matthew Edman
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180
edmanm2@cs.rpi.edu

Paul Syverson
Center for High Assurance Computer Systems
U.S. Naval Research Laboratory
Washington, DC 20375
syverson@itd.nrl.navy.mil

ABSTRACT

Tor is an anonymous communications network with thousands of router nodes worldwide. An intuition reflected in much of the literature on anonymous communications is that, as an anonymity network grows, it becomes more secure against a given observer because the observer will see less of the network. In particular, as the Tor network grows from volunteers operating relays all over the world, it becomes less and less likely for a single autonomous system (AS) to be able to observe both ends of an anonymous connection. Yet, as the network continues to grow significantly, no analysis has been done to determine if this intuition is correct. Further, modifications to Tor's path selection algorithm to help clients avoid an AS-level observer have not been proposed and analyzed.

Five years ago a previous study examined the AS-level threat against client and destination addresses chosen a priori to be likely or interesting to examine. Using an AS-level path inference algorithm with improved accuracy, more extensive Internet routing data, and, most importantly, a model of typical Tor client AS-level sources and destinations based on data gathered from the live network, we demonstrate that the threat of a single AS observing both ends of an anonymous Tor connection is greater than previously thought. We look at the growth of the Tor network over the past five years and show that its explosive growth has had only a small impact on the network's robustness against an AS-level attacker. Finally, we propose and evaluate the effectiveness of some simple, AS-aware path selection algorithms that avoid the computational overhead imposed by full AS-level path inference algorithms. Our results indicate that a novel heuristic we propose is more effective against an AS-level observer than other commonly proposed heuristics for improving location diversity in path selection.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed Applications*

CCS'09, November 9–13, 2009, Chicago, Illinois, USA.
Copyright 2009 ACM 978-1-60558-352-5/09/11 ...$10.00.

General Terms

Algorithms, Design, Measurement, Security

Keywords

Anonymity, autonomous systems, privacy, Tor

1. INTRODUCTION

Much of the existing research into anonymous communication seeks to design and build applications running on top of the existing Internet protocols and infrastructure that allow people to communicate with others without necessarily revealing potentially identifying network information, such as IP addresses. Designs for anonymous communication systems can often be classified into two general categories: high-latency systems and low-latency systems.

High-latency anonymity systems are able to provide better hiding within an anonymity set, but are only practically usable for non-interactive applications that can tolerate delays of several hours or more, such as email. Because of their high-latency, they typically provide stronger anonymity but relative to a smaller set of users. Low-latency anonymity systems, on the other hand, are able to provide better performance and are intended for real-time applications like web browsing. The increased performance, however, often comes at the cost of decreased resilience against certain types of attacks. In particular, low-latency anonymity systems are more easily susceptible to traffic analysis by an adversary who can observe both the connection from a client to the anonymity network and the connection from the network to the client's intended destination [19, 8].

As of August 2009, the most popular publicly deployed low-latency anonymity system is an onion routing network called *Tor* [3]. First gaining public notice in 2004, the Tor network has grown to include over 2,000 volunteer-operated network relays and has an estimated 250,000 or more users. Tor aims to provide anonymity to clients by sending multiply-encrypted data packets through a series of relays distributed across the Internet. Each relay removes a layer of encryption and forwards the result on to either another relay or to the client's intended destination, such as a website.

It is important to keep in mind that the network connections between clients, relays and destinations in the Tor network are rarely (if ever) direct connections. Rather, the Internet is composed of thousands of independent networks called *autonomous systems* (ASes). As data is relayed from the client to a Tor node, it traverses a series of ASes. Previous work [5, 15] has shown that if the same AS appears on the path from the client to the anonymity network and from the anonymity network to the client's destination, then an observer located at that AS can perform a statistical correlation attack to identify the client and her destination.

Intuition from the anonymity literature suggests that as the Tor network grows and volunteers operate relays all over the world, it becomes less likely for a single AS to be able to observe both ends of a connection. Intuition from communications networking is more muddy. On the one hand, there have been great increases in both the size and the geographic diversity of the Tor relay network. On the other hand, this might not be reflected in the number of network providers involved, and corporate and service consolidation could even imply a contraction of the distribution of ASes involved in carrying Tor traffic. In any case, no analysis has been done to determine which intuition is correct. Further, no work has been done to suggest and verify modifications to Tor's path selection algorithm that would help clients avoid an AS-level observer.

In this work, we make the following contributions:

- Based on a more accurate algorithm for inferring AS-level routing paths and a larger set of real-world BGP (Border Gateway Protocol) routing data, we revisit and validate an earlier analysis of the potential threat of AS-level adversaries against the public Tor network.
- We provide a more realistic global model of typical Tor client ASes and destination ASes based on traffic data collected from public Tor relays.
- Using recent Tor directory information and a simulation of Tor's current path selection algorithm, we examine how scalability and performance optimizations made to the Tor software's path selection algorithm have in turn affected its ability to resist AS level attackers.
- We propose and evaluate the effectiveness of some simple, AS-aware path selection algorithms that avoid the computational overhead imposed by full AS-level path inference algorithms.

The rest of this paper is organized as follows. In Section 2, we provide an overview of the Tor design and review the limitations of previous work done to understand its AS-level diversity. Next, in Section 3, we describe the algorithms and data we employed to infer the AS-level paths between Tor clients and their destinations.

We then present in Section 4 the results of a period of data collection that helped to better understand the AS-level distribution of clients and destinations on the public Tor network. Section 5 considers how the growth and evolution of the Tor network over the past five years has affected its susceptibly to an AS-level observer. In Section 6, we propose and evaluate alternative "AS-aware" path selection algorithms that attempt to reduce the probability of a single AS observing both ends of a Tor connection. Finally, in Section 7 we discuss the conclusions we made based on the results of our experiments.

2. BACKGROUND

We first review the design and current implementation of Tor, as well as previous efforts to study the location diversity of the Tor network.

2.1 Tor

Tor [3] is a low-latency anonymity network loosely based on the original onion routing design [7] but with several modifications and improvements over the original design in terms of security, efficiency, and deployability.

The Tor network includes a small set of trusted *authoritative directory servers* responsible for aggregating and distributing signed information about known routers in the network. The signed directory information is also mirrored by other servers in the network. Tor clients periodically fetch the directory information from directory mirrors in order to learn information about other servers in the network, such as their IP addresses, public keys, etc.

To build an anonymous connection through the Tor network, a client first selects an ordered sequence of (usually) three servers. The client then negotiates session keys with each server starting with the first node in the sequence, called the *entry node*. The client can then connect to the middle node via the encrypted tunnel established with the entry node, and then again with the last node in the circuit. The last node is called the *exit node* since it is responsible for establishing the connection from the Tor network to the client's intended destination. The resulting encrypted tunnel through the Tor network is called a *circuit*.

The method the Tor software uses to select the nodes in a client's circuit has undergone many changes since the design was first published. For example, originally clients selected all nodes for their circuit uniformly at random. Later, a primitive form of load-balancing was added wherein relays are selected proportional to a self-reported bandwidth estimate based on how much traffic each server has relayed during a measurement interval. We discuss in greater detail this and other changes made to Tor's path selection algorithm in Section 5.

2.2 Location Diversity

Feamster and Dingledine [5] conducted an empirical analysis of the threat AS-level adversaries could pose to the Tor and Mixmaster [14] networks. The authors defined a *location independence* metric intended to reflect the probability that connections to and from the anonymity network will traverse the same AS. They found that a single AS could observe both ends of a connection 10% to 30% of the time. We note that improved AS path inference algorithms (discussed further in Section 3) have been put forth since the authors' analysis was published. We will later consider how the improved inference algorithms affect the previously published results.

It is not at all clear, however, that their results are applicable to the current Tor network. First, it was conducted at a time when the Tor network consisted of only 33 relays. Since then, the network has seen considerable growth and the number of relays has increased by almost two orders of magnitude. Second, the authors based their analysis on their personal impressions of websites they thought Tor users were likely to be concerned about visiting anonymously (e.g., `indymedia.org`), and only considered clients located at a handful of consumer ISPs within the United States. Third, the authors' simulation only considered Tor clients that chose relays for their circuits uniformly at random. As mentioned above, this is indeed no longer true, yet it has never been investigated whether such changes have had a measurable effect on Tor's susceptibility to an AS-level observer.

Murdoch and Zieliński [15] further considered the threat of an adversary with access to an Internet Exchange (IX) point. An IX is a shared physical location at which multiple ASes are able to interconnect and exchange network traffic with each other. An adversary located at an IX is thus, in theory, able to monitor traffic passing through any of the ASes co-located at the IX. The authors argued that some IXes often sample the network traffic flowing through them for performance analysis purposes. They showed that it is quite possible for a modestly equipped attacker at the IX to perform a traffic analysis attack on the sampled network data and correlate an anonymous sender with her destination, even under limited sampling intervals [15].

To estimate the impact of IX-level observers on the Tor network, Murdoch and Zieliński collected `traceroute` results from volunteers operating Tor relays in the UK. The `traceroute` destinations used were the same list of websites and consumer ISPs used in [5]. The paths returned showed that large IXes, like LINX in England, DE-CIX in Germany, and AMS-IX in the Netherlands, occurred on 32% of the paths collected.

Since Murdoch and Zieliński's analysis used the same list of suspected client origins and destination websites from [5], it is not evident from either study that the results are applicable to the Tor network as a whole. In order to better understand the true nature of where typical Tor clients and servers are geographically located, McCoy et al. [12] collected traffic statistics from a relay they briefly operated on the public Tor network. Their analysis only provided country-level information though. We, however, require both client and destination statistics at the AS level to be able to accurately compute the likelihood of a single AS-level observer monitoring both ends of a Tor connection.

3. AS-LEVEL PATH INFERENCE

Perhaps the most straightforward approach to detecting an AS that exists on both ends of Tor connection would be to use a `traceroute`-like tool to learn the network path from the client to its chosen entry node and the path from the exit node to the destination. There are several problems with such an approach, though. First, since Tor is a volunteer network, we do not have access to all of the relays and thus are unable to run `traceroute` from each machine. There do exist web interfaces that let anyone run `traceroute` queries from special servers across the Internet. These are not sufficient for our purposes because not all Tor relays are located on an AS with a publicly available `traceroute` server. Second, the forward and reverse paths between two hosts on the Internet are often asymmetric. Thus, even if we could run `traceroute` from every Tor relay to a particular destination, we would not be able to do the same in the reverse direction.

Thus, to determine whether an AS exists on both ends of an anonymous connection, we must attempt to predict the path network traffic will take between clients and destinations given an unavoidably incomplete knowledge of the Internet's topology. In the remainder of this section, we describe the AS-level path inference algorithm and corresponding implementation used for the analysis in this paper. We also compare Feamster and Dingledine's [5] results to those we obtained using an improved inference algorithm that has been developed since their paper's publication.

3.1 Path Inference Algorithms

If we could collect routing tables—sometimes referred to as Routing Information Bases (RIBs)—from every AS on the Internet, then determining AS-level paths without `traceroute` would be relatively straightforward; however, this is clearly infeasible for many reasons. Instead, we must make some inferences about AS-level paths given routing information from a subset of ASes. Inferring network paths between two endpoints on the Internet given only partial routing information has been the focus of active research within the networking community over the past decade [10, 9, 18].

Gao previously observed that AS paths typically satisfy a *valley-free* property [6]. Consider customer-provider edges as "uphill" path segments, provider-customer edges as "downhill" segments, and peer-to-peer or sibling-to-sibling edges as "flat" segments. (See next paragraph for meaning of 'peer' vs. 'sibling'.) Thus, in a valley-free AS path, a provider-to-customer edge is followed only by other provider-to-customer or sibling-to-sibling edges. Similarly, a peer-to-peer edge is followed only by provider-to-customer or sibling-to-sibling edges. AS path inference algorithms can then use this heuristic to reduce the number of possible AS paths between two endpoints to only those that satisfy the valley-free property.

Unfortunately, the nature of contractual relationships between ASes are often kept as confidential business information. Thus, we are forced to also infer the relationships between ASes. Gao proposed an algorithm that exploits the valley-free heuristic to infer relationships between ASes [6]. Given one or more RIBs, the algorithm builds an AS-level graph based on which ASes are adjacent to each other in an advertised route. For each route in the routing table, each pair of ASes before the AS with the highest degree in the path is assigned a customer-to-provider relationship. Each pair of ASes after the highest degree AS is assigned a provider-to-customer relationship. Two ASes marked as customers of each other are designated as having a peer-to-peer relationship. If more than some constant number of routes infer that two ASes provide transit for each other, then those two ASes are assigned a sibling-to-sibling relationship.

After inferring AS adjacencies and their relationship types from known routing tables, we have a graph $G = (V, E)$ where V is the set of ASes and E is the set of AS relationships between them. We can then use this graph topology to infer AS paths. Feamster and Dingledine's study of location diversity in the Tor and Mixmaster networks employed Mao et al.'s AS path inference algorithm [9]. Their approach performs a Floyd-Warshall-like all-pairs "shortest policy paths" computation on the graph G. The result of the algorithm is the shortest paths between all pairs of ASes that satisfy the valley-free property.

Note that known routing information is only used for constructing the AS-level topology and inferring relationships. Mao et al.'s algorithm does not use known AS paths for subsequently inferring the path between two ASes. Qiu instead proposed an improved AS path inference algorithm that takes advantage of known AS paths resulting in greater inference accuracy [18]. Known AS paths contained in the routing tables are considered "sure" paths. The ASes with sure paths are called "base" ASes. From the known AS paths, we can also derive other sure paths not explicitly contained in the routing table. For example, consider the known route $R_k = \{v_k v_{k-1} \ldots v_1 p\}$, where v_i is an AS and

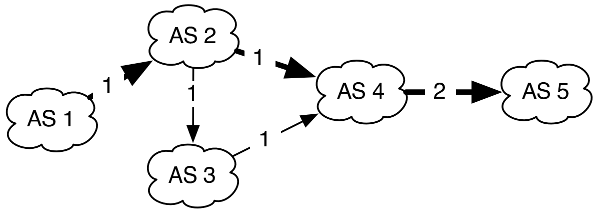

Figure 1: The topology represented above is constructed from three known paths: $\{1, 2, 3\}, \{2, 4, 5\}$ and $\{3, 4, 5\}$. The edge labels indicate the frequency index for each path and sub-path. As we can see, the sub-path $\{4, 5\}$ appears twice in the known AS paths and thus has a frequency index of 2. The bold line indicates an extended path $\{1, 2, 4, 5\}$ obtained by extending path $\{2, 4, 5\}$ by one hop. The resulting path has an uncertain length of 1.

p is a destination prefix. Based on the propagation of route updates via BGP, we know that R_k is derived from the best path $R_{k-1} = \{v_{k-1}, \ldots v_1 p\}$ from AS v_{k-1} to prefix p. Thus, from R_k, we can extract $(k - 1)$ sure paths from a known path of length k.

Since path segments may appear multiple times in both the known AS paths and the derived sure paths, the algorithm maintains a frequency index for each path. The intuition is that paths and sub-paths appearing many times in known AS paths should be preferred in inferred AS paths over those appearing less frequently. Qiu's algorithm infers paths from a source AS to a destination prefix that are *not* contained in the set of sure paths by extending a sure path to adjacent ASes one hop at a time. If the extended path is valley-free, it is added to the routing table. An example of inferring an extended path from one or more sure paths is given in Figure 1. The number of hops a path is extended from a known sure path is called its *uncertain length*. When there are multiple possible extended paths, the algorithm favors inferred paths that have a shorter uncertain length, higher frequency, and shorter overall length.

Qiu showed that an inference algorithm that exploits known AS path information results in a 25% to 27% improvement over Mao et al.'s in terms of inferring precisely the actual route between a source AS and destination prefix.

3.2 Implementation

We developed a multithreaded implementation of Qiu's AS path inference algorithm in C. Full details on Qiu's AS path inference algorithm can be found in [18]. For inferring relationships between ASes, we use Gao's relationship inference algorithm [6].

For input to the algorithm we used RIBs collected by the University of Oregon's RouteViews project [16]. The RouteViews project peers with various ASes and periodically archives routing table dumps and BGP updates. The particular dumps we use throughout the remainder of this paper are RIBs from the OIX, Equinix, PAIX, KIXP, LINX and DIXIE IXes. These routing tables combined represent around 278,000 prefixes, 15.7 million paths, 29,000 ASes and 132,000 edges.

We first examined if the more accurate inference algorithm would affect Feamster and Dingledine's previous analysis of the Tor network's location diversity as it stood in mid-2004. We repeated their experiment using the same list of sender ASes, destination ASes and Tor relays as described in [5]. The only factor that changed was the use of Qiu's more accurate inference algorithm with more comprehensive input routing table data.

The median location independence using the original algorithm was 0.14 in the forward direction and 0.12 in the reverse direction for the sources and destinations given in these tables with the Tor network as it existed at the time. Using the improved algorithm and making no other changes these jump to 0.21 and 0.25 respectively. Overall, a single AS was able to observe either the forward or reverse paths (or both) for 39.4% of the randomly generated circuits–greater than the 10-30% suggested by Feamster and Dingledine. Our results indicate that the problem of an AS-level adversary observing both ends of a connection through the Tor network in 2004 was thus moderately greater than originally thought.

Though understanding path inference is important to understanding our work, and developments purely based on path inference do appear to have a modest impact, our primary focus is elsewhere: determining what the actual distribution is for source and destination ASes of Tor traffic, examining the effect of Tor's tremendous network growth on location independence, and similarly for its path-selection algorithm (which significantly deviates from uniform-at-random for various reasons), and finally, considering novel AS-aware path selection algorithms. It is these questions that we will explore further in the remainder of this paper.

4. SENDER AND RECIPIENT ASES

The senders and recipients used in [5] and [15] were based purely on conjecture. We conducted a real-world study to better understand the true distribution of client origin ASes and destination ASes in the Tor network. We present the results of the study in this section, which were also used for our experimentation described in Sections 5 and 6.

4.1 Data Collection

We ran two Tor nodes on the network of Rensselaer Polytechnic Institute for a period of one week in early September 2008. The first one ran with the default Tor exit policy that blocks a few ports associated with SMTP, file sharing, etc. but otherwise allows traffic to exit the Tor network from it. For collecting statistics on client origin ASes, it is important to note that modern Tor clients will only choose relatively long-running and high-bandwidth nodes designated with a special `Guard` flag by the authoritative directory servers as the first relay in their circuit. Tor clients also tend to choose non-exit relays as the first hop, in order to preserve the somewhat limited available exit node bandwidth. For this reason, the second node did not allow any exiting connections. We also had to wait until the non-exit node attained the `Guard` flag before commencing our actual measurements.

We aggregated the number of connections per AS of clients entering the guard node, as well as the number of connections per destination AS for clients exiting from our exit node. The top fifteen client origin ASes and destination ASes are listed in Table 1. We will discuss our results presently, but we first comment a bit on our data collection method.

	Client ASes				Destination ASes			
	#	AS	CC	Description	#	AS	CC	Description
1	2238	3320	DE	Deutsche Telekom AG	5203	4134	CN	ChinaNet
2	701	4134	CN	ChinaNet	4960	15169	US	Google Inc.
3	672	3209	EU	Arcor	3527	43350	NL	NForce Entertainment
4	576	3269	EU	Telecom Italia	2824	3462	TW	HiNet
5	566	13184	DE	HanseNet Telekommunikation	2085	1668	US	AOL
6	429	6805	DE	Telefonica Deutschland	2029	21844	US	ThePlanet.com.
7	280	12322	FR	Proxad	1530	4837	CN	CNC Group Backbone
8	279	7132	US	AT&T Internet Services	1104	4808	CN	CNC Group Beijing Province
9	276	4837	CN	CNC Group Backbone	1083	3356	US	Level 3 Communications
10	272	9121	TR	TTnet	1011	16265	NL	LeaseWeb
11	251	19262	US	Verizon Internet Services Inc.	979	23393	US	ISPrime, Inc.
12	245	5430	EU	Freenet CityLine GmbH	975	4812	CN	China Telecom
13	230	3215	EU	France Telecom	905	4713	JP	NTT Communications Corp.
14	229	8881	DE	Versatel Deutschland	857	36351	US	SoftLayer Technologies Inc.
15	188	4808	CN	CNC Group Beijing Province	841	26134	US	VeriSign

Table 1: Top 15 Tor client origin and destination ASes observed during a one week period in September 2008. For both origin and destination, less than two percent of the ASes recorded accounted for over half of the connections.

Obviously client privacy is a significant concern, especially for users of the Tor network. Several factors help prevent anyone, including ourselves from learning anything about individual behavior from the data we gathered and present. First, note that the current Tor route-selection policy will not allow circuits containing entry and exit nodes from the same /16 subnet of IP space. Since our two nodes ran from the same IP address it would not be possible to end-to-end correlate any connections using just these two nodes, even if we had collected sufficient data, which we did not. This also means that for any circuit for which we recorded an origin AS, we were technically unable to collect the corresponding destination AS or vice versa. Second, we only used the IP address (origin or destination), nothing else about a given connection such as its time, duration, protocol (for destination connections), number of bytes transmitted, etc. Third, we used IP address only to determine AS. The IP address itself was not recorded. Fourth, we recorded only the aggregate totals for origination ASes and destination ASes rather than an ordered list of those ASes. Thus we did not record even the order of ASes relative to each other.

4.2 Results

We found 2251 distinct ASes for 20638 client connections. Of these, more than half of the ASes produced only a single connection and 85% produced fewer than ten connections. Nearly 43% percent of the connections came from the top twenty five observed client ASes–that is, just over one percent of all observed client ASes. Thus a relatively small percentage of ASes were responsible for the vast majority of the client traffic. Similarly there were 4203 destination ASes recorded during that week for 116781 destination connections. Of these, 34% had only a single connection and 72% had fewer than ten connections. For both origin and destination, less than two percent of the ASes recorded accounted for over half of the connections.

Perhaps the most significant observation from these numbers is that AS-level adversaries are a major unavoidable threat to a large percentage of current Tor usage no matter how route selection might be changed to take into account AS path from clients to entry nodes and from exit nodes to destinations. Controlling a small number of ASes will permit end-to-end correlation on many connections regardless of where the Tor nodes are placed. For example, AS 4134 (ChinaNet) is the number two AS for originating clients and the number one AS for destinations. Without even considering cooperative or business relations with other ASes, this AS passed 3.4% of observed connections from clients and 4.5% of observed connections to destinations. We have no way of knowing the number of client-destination pairs that reside on this one AS. Note also that while AS 3320 (Deutsche Telekom AG) harbored just under half a percent of destination traffic, this single AS was the source of nearly 11% of all connections we observed.

It is instructive to compare the AS information we gathered with that of Feamster and Dingledine [5]. We will address growth of the Tor network in the next section and addressed path inference above. Here we focus only on how well the guesses and data in these papers reflect what we have observed. All the URLs of destinations chosen by Feamster and Dingledine still exist. Nonetheless, we examined the origin AS of 14 out of their 15 destinations and found that eight of these had changed. Thus, of the destinations they chose as likely based on a combination of highly rated Internet properties and their own expectations of likely sites to be of interest to users of anonymity networks, only AS 15169 (Google, rank 2) is in the top 15 we observed. However, when one accounts for the changes in AS, then at least a few more of their entities appear. For example, AS 1668 (AOL, rank 5), is on the list as is AS 3356 (Hotmail, rank 9). Indymedia is on Feamster and Dingledine's list, and its current AS (Savvis, rank 33) appears just below the top 25. Since Savvis is a large IT provider, it is unlikely that this is due solely or even largely to Indymedia; although we have no data to determine this either way.

Still, most of the destinations in our top 15 are not in Feamster and Dingledine's list at all. Their list reflected only ASes in the United States and several, including the first, are not in the United States. Of the nine that are in the U.S., only the three noted above are on their list.

Feamster and Dingledine were not attempting to guess the actual distribution of destinations. They were simply using a plausible list of sites to illustrate the significance of location independence. Some of the sites were known to be very popular in general and others were sites that they thought people concerned with anonymity might care about. In any case, whether or not it would have been an accurate guess in 2004, the actual Tor destination distribution in 2008 is clearly very different from their 2004 list.

We see similar results on the origination side. The Feamster and Dingledine list from 2004 was based on a list of cable and DSL modem providers in the United States. It does not reflect the Tor usage we observed in 2008. There were only three U.S. source ASes in our top 25. Although two of these were reflected on their list of choices in one way or another, the only U.S. source AS at all in the top ten of those we observed is AS 7132 (AT&T Internet Services, rank 8), which Feamster and Dingledine did not choose. The only AS of the eleven they chose that appears in the top 25 is AS 22773 (Cox Communications, rank 24); the only other U.S. origination AS was Verizon (AS 19262, rank 11). Feamster and Dingledine chose Verizon with AS 6995. Although this is still owned by Verizon, it did not appear at all in any of the thousands of source or destination ASes we gathered; thus Verizon has probably changed its use of ASes it owns.

Not surprisingly, our results were more consistent with those of McCoy et al. [12], who also ran Tor nodes within the last year and gathered source and destination information from them. There were important differences between our study and theirs, however. One difference is that they gathered information about destination protocols and traffic amounts rather than destination locations in any sense, while we intentionally avoided gathering any information about specific connections and only aggregated source or destination AS information. They did gather source location information, but their published results are at the level of country rather than AS. Though interesting for understanding who uses Tor, we require AS-level knowledge to analyze and make recommendations concerning an AS-level threat. Their published results for source location also reflect data gathered for only a single day (vs. a week for us). Despite these differences, our results are largely consistent where they overlap. Both studies show the largest number of originations in Germany, far ahead of all other sources. And both have essentially the same order of origin AS/country for the top several sources.

5. TOR GROWS UP

The Tor network and software implementation have undergone many changes since its public deployment in 2004. In particular, the network has grown to include around 1,500 running relays at any given time. The Tor software's path selection algorithm has also undergone many changes to improve the network's load balancing, performance and security. In this section, we seek to better understand how these changes have impacted the Tor network's ability to avoid an AS-level observer.

5.1 Network Growth

To evaluate how only the growth of the Tor network has affected its ability to resist an AS-level observer, we first compare the location diversity of the network as it existed in June 2004 (33 relays) versus September 2008 using the same set of senders and recipients as described in [5]. To avoid skewing our results, we used three different snapshots of the Tor network in September 2008—taken on the 2nd at 0000UTC, the 15th at 0800UTC, and the 29th at 1600UTC—and then averaged the results. We noted above that some of the source and destination ASes used in Feamster and Dingledine's 2004 analysis have since relocated to other ASes. Consequently, we restrict our comparison to only those endpoints that have not moved to another AS since 2004. The resulting comparison is given in Table 2.

From 2004 to 2008 there is a drop in the median probability of a single AS observing both ends of a connection, from .24 to .13 in the forward direction and from .27 to .14 in the reverse direction. The mean probability overall decreased from .38 to .22. That there is a non-trivial decrease is certainly good news. But closer inspection shows this to be a disappointing result for Tor's resistance to such threats. The first thing to observe is how small a drop it is. Simply growing the network has had only a slight effect on AS-level adversaries. This effect should further diminish as the network grows, both because nodes are increasingly more likely to occupy already occupied ASes rather than new ones, and because, even if they do, the more unusual and remote a newly occupied AS is, the greater the number of hops over more common ASes necessary to connect them to clients or destinations.

The more surprising caution on this somewhat positive result is that it is not entirely uniform. Twenty three percent of source/destination AS pairs from the Feamster-Dingledine set had the location independence for forward paths *decrease* as the network grew by two orders of magnitude. Indeed, 12.5% of these AS path combinations were worse off after growth than before. Some of the issues raised above may have played a role. In any case, even substantial network growth does not guarantee improved path diversity: a significant fraction of paths got worse rather than better.

5.2 Path Selection

Tor's path selection algorithm has undergone many changes since its initial deployment. These changes have often been intended to improve client performance, reliability and network load balancing. Sometimes the changes have also been in response to published attacks on the network [17, 1]. We consider the following significant path selection algorithm modifications that have been implemented in Tor and how they might also affect the resistance the Tor software provides clients against AS-level observers.

Weighted Node Selection. Tor clients initially selected all nodes in their path uniformly at random; however, given Tor's volunteer-driven network, clearly not all nodes are able (or willing) to push the same amount of traffic. As a form of primitive load balancing, Tor servers periodically report how much traffic they have relayed and report this information to the directory authorities. Clients receive this information when they download a Tor directory, and then weight their node selection proportional to the amount of bandwidth each node advertises it can handle.

Feamster and Dingledine previously suggested that a good strategy for deploying servers in an anonymity network might be to place them at Tier 1 ISPs that have a high degree of inter-AS connectivity. While Tor can't dictate where its volunteers operate their servers, it is reasonable to believe

	June 2004 (33 relays)						**September 2008** (1239–1303 relays)					
Sender	2914	11643	12182	15130	15169	26101	2914	11643	12182	15130	15169	26101
209	0.49	0.45	0.40	0.39	0.19	0.30	0.17	0.26	0.19	0.51	0.23	0.25
1668	0.39	0.24	0.30	0.30	0.19	0.32	0.18	0.23	0.20	0.25	0.13	0.16
4355	0.38	0.27	0.28	0.27	0.43	0.51	0.13	0.29	0.12	0.20	0.19	0.14
6079	0.62	0.45	0.48	0.24	0.43	0.71	0.12	0.30	0.15	0.22	0.20	0.17
18566	0.39	0.42	0.41	0.32	0.56	0.73	0.18	0.36	0.20	0.31	0.20	0.16
22773	0.56	0.35	0.37	0.21	0.34	0.54	0.21	0.14	0.20	0.20	0.17	0.19
22909	0.21	0.24	0.26	0.22	0.22	0.37	0.19	0.30	0.24	0.25	0.21	0.19
23504	0.39	0.29	0.37	0.33	0.42	0.54	0.49	0.22	0.23	0.19	0.16	0.12

Table 2: Location independence comparison between the Tor network in June 2004 versus an average of three days in September 2008. Despite over 1,000 new relays being added to the network, the mean probability of a single AS observing both ends of a circuit in either the forward or reverse direction only decreased from 37.74% to 21.86%.

that servers operated at Tier 1 ISPs have greater bandwidth available to them than nodes operated by users on, say, a consumer broadband connection. Thus, it is not unreasonable to expect nodes in Tier 1 ISPs to be used more frequently than if all nodes were chosen uniformly at random, which may in turn actually help increase Tor's location diversity.

Distinct /16 Subnets. An easy attack on the Tor network would be for an adversary to simply run two relays on the same machine or network. Eventually a client will pick the attacker's nodes for their entry and exit nodes, potentially allowing the adversary to correlate the sender with her destination. In order to avoid this basic form of Sybil attack [4], Tor clients ensure that the IP address of each node in their circuit is from a different /16 subnet.

Entry Guards. Current Tor clients always pick the first node in their path from a small set of trusted relays called *entry guards* [17]. When a client first runs, it selects a handful of entry guards from available high-bandwidth, high-uptime relays in the network. As previously mentioned, these high-bandwidth nodes may be more likely to exist in highly-connected ASes.

We implemented a simulation of Tor's path selection algorithm based on the TorFlow[1] Python library in order to evaluate the cumulative effect of the above changes to Tor's path selection algorithm on the likelihood of choosing a path that can be observed at both ends by an AS observer. As shown in Section 4, the hypothesized model of typical Tor client and destination ASes does not fit well with the current network usage. Instead, we used the distribution of client ASes and destination ASes we collected from a public Tor relay.

Using the Tor path simulator and the same three snapshots of the Tor network from Section 5.1, we generated 15,000 paths—5,000 for each snapshot. We also generated 15,000 paths (again, 5,000 for each Tor directory snapshot) using where entry and exit nodes were selected uniformly at random to represent how a Tor client from 2004 would choose paths.

Sender and recipient ASes were selected proportional to their observed distribution on the public Tor network. We then used our AS path inference implementation and archived RouteViews BGP data corresponding to each snapshot to infer the forward and reverse paths between senders and entry nodes, and exit nodes and destinations, resulting in a total of 60,000 AS paths to infer. The following are the aggregated results over all three snapshots:

	Forward	Reverse	**Total**
Uniform	12.79%	13.23%	20.49%
Weighted (Tor)	10.92%	11.14%	17.81%

The first row of the above table gives the probability of an AS observing both ends of a connection for a uniformly random node selection. The second row gives the same results but instead for Tor's current path selection algorithm, incorporating bandwidth weighting, entry guards and distinct /16 subnet enforcement. Even though the algorithm Tor uses to select relays in path was done primarily for performance reasons, we see that Tor's path selection algorithm has also had a small but positive and non-negligible impact on the probability that a single AS will be able to observe both ends of a typical client's connection. We stress that the same Tor directory information, sender and recipient distributions, and routing table data were used for both experiments. The only difference between the two was the method used for choosing entry and exit nodes.

5.3 Effectiveness of Distinct /16 Subnets

Tor's policy of ensuring that every node in a circuit is selected from a distinct /16 subnet seems like a reasonably effective approach to increasing AS-level diversity within a circuit. We wanted to investigate how effective this practice actually is on the current Tor network. Taking a snapshot of the Tor network in mid-September 2008, we observed 1238 running relays existing in a total of only 474 different ASes. Of those 1238 relays, 417 of them had an IP address in the same /16 subnet of another Tor relay. More surprisingly, a total of 876 relays (or about 70%) existed in the same AS as at least one other relay but had a different /16 network address from it. Such pairs of relays would not be detected by Tor's distinct /16 subnet enforcement. Of those 876 relays, 850 not only had a distinct /16 but also a distinct /8 network address.

In order to see how often such nodes appear in the entry and exit positions of the same circuit, we again generated 15,000 paths according to Tor's path selection algorithm, including the requirement that nodes belong to distinct /16 subnets. We then resolved the entry and exit node IP ad-

[1] `https://svn.torproject.org/svn/torflow/`

dress to their origin ASes. Out of 15,000 paths, 113 (approximately 1 out of every 133 circuits) contained an entry and exit node that resided in the same AS despite having an IP address from different /16 subnets. Within those 113 paths, all but four also had a distinct /8 network address.

These results suggest that Tor's policy of requiring nodes in a path to have IP address in distinct /16 subnets is largely effective, though may not be stringent enough. Increasing the policy to enforcing distinct /8 subnets appears be a reasonable suggestion, but is by no means a solution to avoiding an AS-level observer.

6. AS-AWARE PATH SELECTION

Based on the results above, it is apparent that simply increasing the size of the Tor network with volunteer-operated relays is not a sufficient approach to significantly reducing the threat of an AS-level observer. Rather, a more proactive approach on the part of Tor clients is needed. In this section, we evaluate the effectiveness of various modifications to Tor's path selection algorithm that also try to enforce better AS-level diversity.

6.1 Using diversity within the Tor network

Diversity of relay locations within the Tor network has often been considered a boon to anonymity. Tor Project Proposal 144 [13] specifically suggests requiring that different nodes in a circuit not only exist in disjoint /16 networks, but also come from different ASes.[2] Another potential method to enforce location diversity in Tor circuits is to ensure each hop in a client's circuit is located in a different country. If effective, this would be an appealing option since the Tor software recently started including a database in many of its binary distributions for mapping IP addresses to countries thus making implementation quite simple.

But how effective are these various diversity proposals against an AS level adversary? Extending our Tor path simulator used in Section 5, we experimented with adding the requirement to Tor's existing path selection algorithm that each node in a circuit be located in a different country. Second, instead of requiring unique countries, we ensured that each node in a circuit exists in a different AS. Table 3 compares the results of the country-aware (Unique-CC) and AS-aware (Unique-AS) algorithms versus Tor's current algorithm and selecting nodes uniformly at random.

While the two simple approaches result in a moderate decrease in the probability that an AS will exist on both ends of a connection, the results are not as striking as proponents of Proposal 144 would suggest. Perhaps the most interesting point to note is that there is effectively little difference between choosing nodes from distinct ASes versus choosing them from distinct countries. Given that the latter can more easily be accomplished with little change to the Tor software, it does not seem worthwhile to pursue adding mechanisms by which clients can reliably and securely determine the origin AS for all Tor relays.

6.2 Approximating AS Paths

The previously discussed Unique-AS and Unique-CC path selection algorithms only consider properties of the nodes themselves when constructing a path, but did not offer much of an improvement in terms of resilience against an AS level observer. Unfortunately, the AS path inference algorithms described earlier in this paper and in the networking literature are expensive both in terms of computational complexity and storage requirements. To give the reader an idea of the space required, the six routing table dumps used for the analysis in this paper occupied around 1.47 GB of disk space uncompressed and several hundred megabytes compressed. For Tor clients on connections with moderate bandwidth, distributing full routing information is clearly not practical.

	Forward	Reverse	**Total**
Uniform	12.79%	13.23%	20.49%
Weighted (Tor)	10.92%	11.14%	17.81%
Unique-CC	10.41%	11.24%	17.61%
Unique-AS	10.07%	10.14%	16.73%
Approx. AS Path ($n = 1$)	*6.29%*	*6.01%*	*11.09%*
Approx. AS Path ($n = 3$)	*3.17%*	*3.34%*	*6.23%*

Table 3: Percentage of circuits generated by current and proposed Tor path selection algorithms that result in a single AS being able to observe both sides of the connection in either the inferred forward or reverse AS-level paths. The approximate AS path heuristic we propose yields the most effective avoidance of AS level observers.

Instead, we consider a more practical approach wherein the handful of trusted and likely more capable Tor directory authorities generate a smaller approximation of the Internet's global AS structure and distribute only this "snapshot" to clients. Given a snapshot of the AS-level topology, we show that clients can apply some simple and efficient heuristics in order to approximate the sequence of ASes a packet would traverse on entry to or exit from the Tor network.

6.2.1 *Generating the AS Topology Snapshot*

The first step in this approach proceeds in much the same manner as the full AS path inference algorithm. One or more of the Tor directory authorities fetch a set of RIBs and construct a directed graph where the vertices are ASes and the edges are interdomain routing connections between ASes. Each edge is labeled with the type of AS relationship shared between the two endpoints as inferred by any of the well known relationship inference algorithms [6, 2]. Also associated with every edge is a *frequency* value that indicates how many AS paths in the input RIBs contained that edge. The authorities also produce a separate table that maps IP prefixes to the AS or ASes that originate that prefix in the input RIBs, in order for clients to map a destination IP address to an AS number.

6.2.2 *Approximating an AS-level Path*

When a Tor client downloads an initial network consensus and descriptors for all relays contained therein, the client would also download the AS topology and prefix table snapshots computed and agreed upon by the directory authorities. When the client wants to establish a circuit, it chooses an entry and exit relay according to Tor's normal path selection algorithm. We then use the following algorithm for estimating whether there is a potential for an AS-level adversary to observe both ends of the connection.

[2] Tor Project Proposals are intended to provide an open way to evolve Tor specification and design. They are *very* roughly similar to IETF/IRTF RFCs in this respect [11].

First, the IP addresses of the client, chosen entry and exit nodes, and destination are mapped to their origin ASes using a longest-matching prefix search in the downloaded prefix table. Using the downloaded topology snapshot and a simple modified implementation of Dijkstra's algorithm, the client then finds *all* shortest forward and reverse AS paths from the client's AS to the entry node, and from the exit node to the destination. The algorithm next discards any paths in the resulting set of shortest paths that do not meet the "valley free" property that is common in Internet routing. Finally, the remaining entry paths and exit are sorted according to the cumulative frequency values for each edge in the path. The shortest n paths with the greatest cumulative edge frequency values are then assumed to be the n most likely AS-level paths from a source AS to the destination.

The above process is repeated for both the forward and reverse entry and exit paths. If the same AS appears in any of the n entry paths and any of the n exit paths, the chosen entry-exit node pair is discarded and a new pair is selected. The AS path approximation algorithm above is repeated until a "safe" entry-exit relay pair is found.

6.2.3 Evaluation

We used the routing tables listed in Section 3.1 to generate the AS topology snapshot and prefix tables. Our proof-of-concept implementation used a simple text-based format for both files. While the input routing data used in our experiments was over 1.47GB, the total size of an AS topology snapshot was only 1.6MB, which was further reduced to 602KB using the same basic `gzip` compression method already used in the Tor software. Using `bzip2`, the topology snapshot was further reduced to around 500KB. The prefix table was much larger at 6.1MB total uncompressed and just over 2MB compressed (or 1.4MB using `bzip2`). Given that Tor clients currently download over 2MB of relay data when bootstrapping, we believe the download requirements imposed by our approach is quite reasonable for a vast majority of Tor clients. Like relay directories, the AS topology and prefix snapshots can be signed and distributed to directory mirrors as well, reducing the load on the directory authorities even further.

In evaluating the effectiveness of our approach, we generated 15,000 paths following the algorithm above using Tor directory information from three days in September 2008. We repeated the experiment twice; once with $n = 1$ and again with $n = 3$. From Table 3, we see that applying our simple heuristics to a condensed AS topology snapshot resulted in a significant drop in probability that a single AS is able to observe both ends of a Tor circuit. Our algorithm, despite being written in Python with no particular optimizations, was also reasonably efficient in our experiments. It took our implementation around 1-3 seconds to choose a safe entry-exit pair for a particular client-destination combination when $n = 1$, and around 3-5 seconds for $n = 3$.

There are also several potential client-side implementation optimizations available with this approach. Tor clients currently pick a small set of entry guards to always use as the first hop in any circuit they construct. The top n approximated AS paths from the client to each of its chosen entry guards could be computed when the client first receives the AS topology snapshot. Later, when the client attempts to establish a circuit, only the exit path needs to be computed.

We finally note that the format of our AS topology and prefix snapshots was extremely straightforward, and little effort was made to further minimize their size. For example, it may be possible to further reduce the size of the prefix table using techniques from the networking research community that have focused on small and efficient longest prefix-matching lookup tables. On the subject of efficiency, it would be an interesting future research question to consider how frequently the directory authorities must refresh their routing table information in order to produce an updated AS-level topology snapshot for clients.

7. CONCLUSIONS

It is important to keep in mind that the network's growth has still been beneficial to Tor clients in other ways, even if it hasn't significantly reduced the threat of AS-level observers. The increased number of nodes improves the number of simultaneous clients the network is able to support. It also decreases the amount of traffic that an inside adversary controlling a given number of nodes is likely to be able to observe.

Similarly, suggestions to require that entry and exit nodes for a given Tor circuit reside in different countries have been motivated at least as much by concern over attacks from administrative or governmental adversaries using legal or extralegal means as by concern about threats from the structure of the underlying communications network. Thus, even if these provide only small improvement over current Tor route selection against AS-level adversaries, they provide other benefits as well.

In this paper we have examined connection over the Tor network in the face of AS-level adversaries. We used a newer more accurate path-inference algorithm vs. that used by Feamster and Dingledine, and showed that, even for the sources, destinations and Tor network they examined, connections were slightly less location independent than they thought. More importantly, we showed via observation of actual usage that the ASes of circuit originators and of circuit destinations of the Tor network in 2008 were significantly different from those mentioned by Feamster and Dingledine. A significant percentage of circuits originate from a small number of ASes. Likewise a significant percentage were intended for destinations at a small number of ASes. Thus, no matter how Tor routing is done, much existing Tor traffic is vulnerable to AS-level adversaries. We showed that /16 separation and even /8 separation for Tor circuits did not imply AS-independence for those circuits. We also showed that the tremendous growth of the Tor network has had only small impact on its AS-level path independence, and a significant percentage of paths had AS-level independence diminish as the network grew.

And, we showed that requiring unique countries for Tor nodes was as effective for providing non-intersecting AS paths as was requiring unique ASes. Thus, the easier to implement and independently motivated country separation is more sensible. Though country separation was the easiest and lowest overhead approach, by far the most effective heuristic approach to path independence (as measured by general percentage of circuits compromised) was the path approximation approach.

We also considered a heuristic of choosing shortest AS paths to entry guards at the entry side of a circuit as an easy to implement and apply technique to increase AS independence. As an approach to reducing the average chance of AS-level path compromise it faired worse than country indepedendence (in the reverse direction) and much worse than AS path approximation. But it might be the basis of a much better approach for specific clients even if not better for the network and userbase as a whole. For clients that typically don't change AS much, if, e.g., one could find or set up guard nodes within the client's home AS, then that client would be immune to AS-level attack except by any but his local AS. If such a set-up is not feasible, then doing the same within as few AS-level hops as possible would still have clear advantage for the given client.

An interesting direction for future work would be to incorporate out-of-band knowledge about which ASes are colocated at various IXes. We would then be able to better estimate the impact of IX-level observers and compare it to our results that consider only AS-level observers. Another direction would be to determine the impact of AS-aware routing on other threats to anonymity for Tor. For example, currently an initial node tells little about the client, if shortest AS path to entry guards is used to pick entry guards, then a second node or observer of the network link from the first node in a circuit will be able to make reasonable inferences about the AS of the client. Also, given the number of circuits that route from/to a small number of ASes, it may be reasonable to design route selection based on trust that the originator places on the AS(es) between her and the entry node. We intend to explore this further in future work.

Acknowledgments

The authors gratefully acknowledge Peter Palfrader for providing us with the archive of Tor network data used in our analysis.

8. REFERENCES

[1] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. Low-resource routing attacks against Tor. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007)*, Washington, DC, USA, October 2007.

[2] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, and G. Riley. As relationships: Inference and validation. *ACM SIGCOMM Computer Communication Review*, 37:29, 2007.

[3] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[4] J. Douceur. The sybil attack. In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, March 2002.

[5] N. Feamster and R. Dingledine. Location diversity in anonymity networks. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004)*, Washington, DC, USA, October 2004.

[6] L. Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Transactions on Networking*, 9:733–745, 2000.

[7] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Hiding routing information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, May 1996.

[8] B. N. Levine, M. K. Reiter, C. Wang, and M. K. Wright. Timing attacks in low-latency mix-based systems. In A. Juels, editor, *Proceedings of Financial Cryptography (FC '04)*, pages 251–265. Springer-Verlag, LNCS 3110, February 2004.

[9] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang. On as-level path inference. In *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS International conference on Measurement and Modeling of Computer Systems*, pages 339–349, New York, NY, USA, 2005.

[10] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an accurate as-level traceroute tool. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 365–378, 2003.

[11] N. Mathewson. The Tor proposal process. Tor Project Proposal 001, January 2007.

[12] D. McCoy, K. Bauer, D. Grunwald, T. Kohno, and D. Sicker. Shining light in dark places: Understanding the Tor network. In N. Borisov and I. Goldberg, editors, *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, pages 63–76, Leuven, Belgium, July 2008. Springer.

[13] Mfr. Increase the diversity of circuits by detecting nodes belonging the same provider. Tor Project Draft Proposal 144, June 2008.

[14] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster Protocol — Version 2. IETF Internet Draft, July 2003.

[15] S. J. Murdoch and P. Zieliński. Sampled traffic analysis by internet-exchange-level adversaries. In N. Borisov and P. Golle, editors, *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, Ottawa, Canada, June 2007. Springer.

[16] U. of Oregon. RouteViews archive. `http://archive.routeviews.org`, September 2008.

[17] L. Øverlier and P. Syverson. Locating hidden servers. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE CS, May 2006.

[18] J. Qiu and L. Gao. As path inference by exploiting known as paths. In *Proceedings of the 2006 Global Telecommunications Conference*, December 2006.

[19] A. Serjantov and P. Sewell. Passive attack analysis for connection-based anonymity systems. In *Proceedings of ESORICS 2003*, October 2003.