

Measuring and Mitigating AS-level Adversaries Against Tor

Rishab Nithyanand*, Oleksii Starov*, Adva Zair†, Phillipa Gill* and Michael Schapira†

*Stony Brook University

Email: {rnithyanand, ostarov, phillipa}@cs.stonybrook.edu

†Hebrew University of Jerusalem

Email: {adva.zair@mail, schapiram@cs}.huji.ac.il

Abstract—The popularity of Tor as an anonymity system has made it a popular target for a variety of attacks. We focus on traffic correlation attacks, which are no longer solely in the realm of academic research with recent revelations about the NSA and GCHQ actively working to implement them in practice.

Our first contribution is an empirical study that allows us to gain a high fidelity snapshot of the threat of traffic correlation attacks in the wild. We find that up to 40% of all circuits created by Tor are vulnerable to attacks by traffic correlation from Autonomous System (AS)-level adversaries, 42% from colluding AS-level adversaries, and 85% from state-level adversaries. In addition, we find that in some regions (notably, China and Iran) there exist many cases where over 95% of all possible circuits are vulnerable to correlation attacks, emphasizing the need for AS-aware relay-selection.

To mitigate the threat of such attacks, we build Astoria—an AS-aware Tor client. Astoria leverages recent developments in network measurement to perform path-prediction and intelligent relay selection. Astoria reduces the number of vulnerable circuits to 2% against AS-level adversaries, under 5% against colluding AS-level adversaries, and 25% against state-level adversaries. In addition, Astoria load balances across the Tor network so as to not overload any set of relays.

I. INTRODUCTION

Tor is a popular anonymity system for users who wish to access the Internet anonymously or circumvent censorship [15]. The increasing popularity of Tor has recently made it a high-value target for blocking and denial of service [13], [29], [43] and traffic correlation attacks to deanonymize users [24], [25], [30], [31], [37]. Traffic correlation attacks, which correlate traffic entering the Tor network with traffic exiting it, are no longer solely in the realm of academic research with recent revelations about the NSA and GCHQ actively working to implement them in practice, in collusion with Internet Service Providers (ISPs) [3], [5], [7].

Traffic correlation attacks have been shown to be feasible and practical for network-level attackers. Specifically, a traffic

correlation attack may be implemented by any autonomous system (AS) that lies on both the path from the Tor client to the entry relay and on the path from the exit relay to the destination. Previous studies have demonstrated the potential for this type of attack [16], [18], [25]. Proposed defenses include relay selection strategies to avoid ASes that are in a position to launch them [9]. However, recent work [41] has shown that these strategies perform poorly in practice.

The threat of network-level adversaries has been exacerbated by a recent study which highlights that the set of ASes that are in a position to perform traffic correlation analysis is potentially much larger due to asymmetric routing, routing instabilities, and intentional manipulations of the Internet’s routing system [39], [40]. These attacks significantly raise the bar for relay-selection systems. Specifically, they require the relay-selection system be able to accurately measure or predict network paths in both the forward and reverse direction. Measuring the reverse path between two Internet hosts is non-trivial, especially when the client does not have control over the destination, as is commonly the case for popular Web services. While solutions for measuring reverse paths have been proposed [27], they are still not widely deployed or available.

In this paper, we make contributions in two dimensions. First, we quantify the threat posed by these new attacks. Second, we develop a relay selection method to minimize their impact.

Measuring the threat faced by Tor. We leverage up-to-date maps of the Internet’s topology [23] combined with algorithmic simulations [22] to predict which ASes are in a position to perform traffic correlation analysis on forward or reverse paths. We validate this technique and show that it provides a reasonable *estimate* on the threat faced from AS-level attackers. We then augment our analysis with techniques to identify ASes owned by a single organization (sibling ASes) in order to gain a clearer picture of which ASes are likely to collude with each other. This provides a more complete picture of network-level threats than previous work. In addition, we consider the threat from state-level attackers that have insight into traffic transiting through all regional ASes. Through these techniques and our experiments, we make the following key observations:

- Up to 40% of circuits constructed by the current Tor client are vulnerable to network-level attackers.

- Up to 37% of all sites in our study, when loaded from Brazil, China, Germany, Spain, France, England, Iran, Italy, Russia, and the United States had main page requests that were reached via a *vulnerable* path (i.e., a path that contained network-level entities in a position to launch traffic correlation attacks), when loaded by the vanilla Tor client.
- Connections from China were found to be most vulnerable to network-level attackers with up to 86% of all Tor circuits and 56% of all main page requests to sites in the study being vulnerable to colluding network-level attackers.
- For up to 8% of the requests generated from China and Iran, over 95% of *all possible* Tor constructed circuits were vulnerable to correlation attacks by network-level attackers.
- Reducing the number of entry guards can result in an increase in vulnerability of Tor circuits. In particular, we found that using a single guard significantly increases the threat from traffic correlation attacks, while the difference between using two and three guards is marginal.
- State-level attackers are in a position to launch correlation attacks on up to 85% of all Tor constructed circuits.

Mitigating the threat of AS-level adversaries. We propose, construct, and evaluate Astoria— an AS-aware Tor client that includes security and relay bandwidth considerations when creating Tor circuits. Astoria is the first AS-aware Tor client to consider the recently proposed asymmetric correlation attacks [39], [40]. When there are safe alternatives, Astoria actively avoids using circuits on which asymmetric correlation attacks might be launched. It also leverages methods for identifying sibling ASes [10] when determining whether or not a given circuit is safe. In the absence of a safe path, Astoria uses a linear program to minimize the threat posed by any adversary. Finally, Astoria considers the bandwidth capabilities of relays while making AS-aware relay selection decisions. When there are multiple safe relay selections, Astoria aims to be a good network citizen and distributes load across Tor relays in the same manner as the vanilla Tor client. Therefore, in spite of selecting safer relays, Astoria will not overload any single set of relays.

Paper outline. In Section II we briefly overview how the current Tor client performs relay selection and circuit construction, describe the current state of research in relay selection for Tor, and introduce our adversary model. In Section III we describe the components of our measurement toolkit used for detecting network-level attackers on Tor circuits. We then present some interesting results regarding the vulnerability of Tor constructed circuits and the general potential for attack by single AS-, sibling AS-, and state-level attackers. In Section IV, we present the details of our AS-aware client – Astoria. A performance and security evaluation of Astoria is performed in Section V. In Section VI, we discuss the known shortcomings of Astoria and motivate directions for future research on AS-aware clients. We make our conclusions in Section VII.

II. BACKGROUND AND MOTIVATION

We now provide background on Tor relay selection, related work in this area, and our adversary model.

A. Tor relay selection

The Tor anonymity network consists of approximately 6,000 relays (Tor routers). Most requests made through a Tor client are sent to their destination via a three-hop path known as a circuit. Each circuit consists of an entry, middle, and exit relay. The entry-relay communicates directly with the client and the exit-relay communicates with the destination. The fundamental idea is that no single relay in the circuit learns the source and destination.

In its early days, Tor selected relays for each circuit hop uniformly at random from the set of available relays. This was changed in order to improve performance (by preferring to route through higher bandwidth relays [8]) and security [11]. In today’s Tor network, based on certain performance characteristics such as reliability, bandwidth served, and up-time, relays may earn certain flags that make them a preferential choice for various roles during circuit construction.

One such flag is the guard flag. New relays joining the Tor network are monitored for stability and performance via remote measurements for a period of up to eight days [4]. At this point, relays that have demonstrated stability and reliability are assigned a guard flag. Relays with a guard flag earn the ability to serve as the entry-relay to the Tor network. By default the Tor client selects three guards to be used as entry-relays for all circuits for a prolonged period of time. The main ideas behind the selection of a fixed set of entry-relays are (1) to reduce the possibility that a client will select an entry- and exit-relay operated by the same entity (after prolonged use), (2) prevent attacker-owned entry-relays from denying service to clients that are not also using an exit-relay owned by the attacker, and (3) increase the cost to an attacker that wishes to be chosen as an entry-relay, by requiring them to earn the guard flag [4].

In addition to picking relays that are more stable and reliable, for other locations on a circuit, the Tor client also requires that (1) no two routers on a circuit share the same /16 subnet and (2) no routers in the same family (as advertised by the router) may be chosen on the same circuit. [8].

B. Related work

The threat of correlation attacks by AS-level adversaries on the Tor network was first identified and empirically evaluated by Feamster and Dingedine [18] in 2004, when the Tor network had only 33 relays and significantly different relay selection algorithms. The study revealed that 10-30% of all circuits constructed by Tor had a common AS that could observe both ends of the circuit. Shortly after, by constructing efficient traffic correlation attacks while considering network-level adversaries, Murdoch and Danezis [30] and Murdoch and Zielinski [31] demonstrated that the threat from AS-level attackers was one of practical concern. In 2009, Edman and Syverson [16] found that the threat of AS-level adversaries had not reduced since [18], in spite of revised relay selection strategies and substantially larger number of relays in the network.

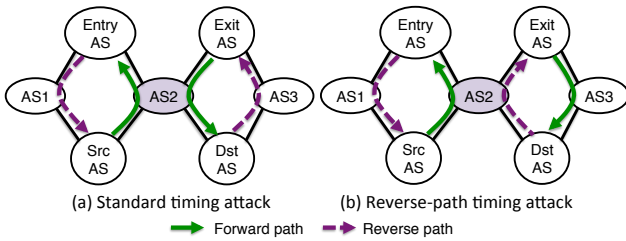


Fig. 1: Standard and reverse-path traffic correlation attacks. In the standard traffic correlation attack, AS2 must observe the direction of the connection that data is flowing on (forward path). In the reverse-path traffic correlation attack AS2 can infer the data flow using ACK numbers on the reverse path.

In addition, Edman and Syverson [16] were the first to consider threats from network-level attackers due to the asymmetric nature of Internet routing. Using the 2009 topology of the Internet, AS paths inferred by Qiu’s algorithm [32], and AS relationships inferred by Gao’s algorithm [20] they found that in their experiments up to 39% of all Tor circuits were vulnerable to network-level adversaries that performed attacks on forward- and reverse-paths. Most recently, Vanbever *et al.* [40] and Sun *et al.* [39], presented RAPTOR, an AS-level attack integrating BGP interception with the first correlation attack that takes advantage of the asymmetric nature of Internet routing, to exactly de-anonymize Tor users with up to 90% accuracy in just 300 seconds. Similarly, Johnson *et al.* [25] performed an empirical evaluation of the effect of network-level adversary bandwidth investment strategies, Tor client location, and Tor client use (*e.g.*, for IRC, browsing, BitTorrent, *etc.*). They found that a network-level adversary could effectively de-anonymize most Tor users within six months with very low bandwidth costs. These works emphasize the need for Tor relay selection strategies to consider ASes that lie both, on the forward- and reverse-paths between the (client, entry) and (exit, destination).

Perhaps most closely related to our work, in terms of end-goals and evaluation methodology, Akhoondi *et al.* [9], constructed LASTor, a Tor client which explicitly considered AS-level attackers and relay locations while constructing Tor circuits. While LASTor appeared to successfully reduce path latencies and the probability of common ASes at either end of the Tor circuits, it neglected the capacity of relays selected by the system. Relay capacity is an important variable to consider to ensure that custom relay selection schemes do not overload a small set of relays, therefore reducing the performance of the entire network. Their evaluation, based on only HTTP HEAD requests (as opposed to complete webpage loads), did not stress the system sufficiently to reveal the issues associated with capacity-agnostic relay selection. Further, LASTor does not consider an adversary that may (1) collude with other ASes or operate at the state-level, and/or (2) only need to be on one of the asymmetric path segments between source and entry-relay; and exit-relay and destination (*e.g.*, RAPTOR).

C. Adversary model

In the standard view of traffic correlation attacks, an AS needs to lie on the forward path¹ between the source and destination (*i.e.*, on the solid green colored path segments in Figure 1 (a)). With this point the adversary (AS 2) can view the packet sizes and timings as transmitted from the source to destination, going-into and coming-out-of the Tor network and directly perform a traffic correlation attack.

However, recent work by Vanbever *et al.* [40] and Sun *et al.* [39] highlights the fact that an adversary on the reverse path may also learn packet size and timing information via the TCP Acknowledgement (ACK) field. Figure 1(b) illustrates this case. AS 2 can directly observe packet timings between the source and entry-relay AS (Entry AS), but can only observe ACKs from the destination back to the exit-relay AS (Exit AS).

In this view, an adversary has the potential to launch a traffic correlation attack on a Tor circuit as long as the following criteria are satisfied:

Let $p_{src \rightarrow entry} = \{AS_1, AS_2, \dots, AS_n\}$ be the set of ASes on the path from the source (Tor client) to the selected entry-relay (this set includes the entry-relay AS), $p_{entry \rightarrow src} = \{AS'_1, AS'_2, \dots, AS'_m\}$ be the set of ASes on the path from the entry-relay back to the source, and $p_{entry \leftrightarrow src} = p_{entry \rightarrow src} \cup p_{src \rightarrow entry}$. We similarly define paths to and from the exit-relay and destination (*e.g.*, a popular content provider, or other Web service) as $p_{exit \rightarrow dst}$, $p_{dst \rightarrow exit}$, and $p_{exit \leftrightarrow dst}$.

We say that a Tor circuit is vulnerable to a traffic correlation attack if there exists an AS A_i such that:

$$A_i \in \{p_{src \leftrightarrow entry} \cap p_{exit \leftrightarrow dst}\} \quad (1)$$

Similar to prior work on relay selection, we assume that our adversary is an autonomous system (AS), or an entity working with the cooperation of ASes (*e.g.*, governments). However, while all previous work only considers the standard view of network attacks, we also consider attackers that may lie on the reverse-path, as described above. In addition, we also include the possibility that some sets of ASes may collude with each other to de-anonymize Tor users. Specifically, we consider that an AS may collude with sibling ASes [10] (*i.e.*, other ASes owned by the same organization) and ASes that may collude with each other on behalf of a state-level adversary. Finally, as part of our relay selection algorithms (Section IV), we consider a probabilistic relay selection strategy that minimizes the amount of traffic that is observable by any single attacker over a period of time.

III. MEASURING ADVERSARY PRESENCE

In this section, we investigate the prevalence of the adversary described in Section II. First, we detail how prediction of AS paths between a source and a destination is performed and how sets of potential attacking ASes are generated. Then we present the experimental methodology used to make these measurements. Finally, we present the results of these experiments.

¹Here we use ‘forward path’ to refer to the direction of data flow in the TCP connection

A. Predicting potential attacker ASes

Adversaries that can exploit asymmetric routing present a challenge to measuring their prevalence. The addition of potential attackers on the reverse-path between a source and destination implies the need for identifying potential attackers (*i.e.*, ASes) on the reverse-paths between the client and entry-relay (and the exit-relay and destination). This poses a challenging measurement problem, since reliably measuring information about reverse-paths is currently not possible. While Reverse Traceroute [27] would be a useful tool for these measurements, it is currently not widely deployed.

Additionally, since our measurement toolkit was assembled with the goal of integration with our Tor client – Astoria (Section IV), using external measurement and control-plane mapping tools was not an option. This is because such tools require knowledge of the clients’ intended destination – an undesirable option for an anonymity tool such as Tor. Thus, any measurement or path prediction needs to be performed on the Tor client without leaking any information to attackers or third party tools and service providers.

To address the challenges of reliably measuring reverse-paths or use control-plane mapping tools, we employ an efficient path prediction approach which leverages up-to-date maps of the AS-level Internet topology [23], and algorithmic simulations that take into account a common model of routing policies [22].

AS-level topology. We perform path prediction using an empirically-derived AS-level Internet topology. In this abstraction, the Internet is represented as a graph with ASes as nodes and edges as connections between them. Connections between ASes are negotiated as business arrangements and are often modeled as two main types of relationship: *customer-provider* where the customer pays the provider for data sent and received; and *settlement-free peering* or *peer-peer* where two ASes agree to transit traffic at no cost [21].

However, in practice AS relationships may violate this simple taxonomy *e.g.*, ASes that agree to provide transit for a subset of prefixes (*partial transit*) or ASes that have different economic arrangements in different geographic regions (*hybrid relationships*) [23]. It can also be the case that two ASes are controlled by the same organization *e.g.*, because of corporate mergers such as Level 3 (AS3356) and Global Crossing (AS3549) or organizations that leverage different AS numbers in different regions such as Verizon (AS701, 702, 703). Additionally, integrating IXPs is a complicated research subject due to a dearth of measurement data to inform how they should be incorporated – *e.g.*, just because two ISPs peer at an IXP does not mean all paths including these ISPs will traverse the IXP. The AS-level topology we leverage takes partial transit and hybrid relationships into account, but ignores IXPs (which would result in a significant over-estimation of our measurements, due to their peering meshes). We use techniques discussed and validated by Anwar *et al.* [10] for detecting sibling ASes. This is done to identify ASes that are likely to collude with each other.

Routing policies. Routing on the AS-graph deviates from simple shortest path routing because ASes route their traffic based on economic considerations. We use a standard model

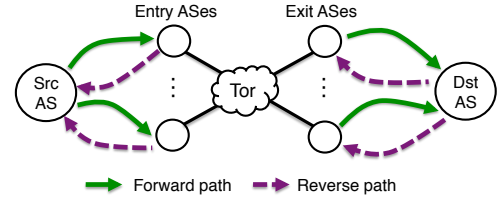


Fig. 2: Illustration of the AS paths that the client needs to predict, note that these paths must be predicted for each potential entry and exit relay in both the forward and reverse direction.

of routing policies proposed by Gao and Rexford [21]. The path selection process can be broken down into the following ordered steps:

- *Local Preference (LP)*. Paths are ranked based on their next hop: customer is chosen over peer which is chosen over provider.
- *Shortest Paths (SP)*. Among the paths with the highest local preference, prefer the shortest ones.
- *Tie Break (TB)*. If there are multiple such paths, node a breaks ties: if b is the next hop on the path, choose the path where hash, $H(a, b)$ is the lowest.²

This standard model of local preference [21] captures the idea that an AS has incentives to prefer routing through a customer (that pays it) over a peer (no money is exchanged) over a provider (that it must pay).

In addition to selecting paths, ASes must determine which paths they will announce to other ASes based on export policies. The standard model of export policies captures the idea that an AS will only load its network with transit traffic if its customer pays it to do so [21]:

- *Export Policy (EP)*. AS b announces a path via AS c to AS a iff at least one of a and c are its customers.

Computing paths following these policies using simulation platforms (*e.g.*, CBGP [33]) can be computationally expensive which limits the scale of analysis. Thus, we employ an algorithmic approach [22] that allows us to compute all paths to a given destination in $\mathcal{O}(|V|+|E|)$ where $|V|$ is the number of ASes and $|E|$ is the number of edges.

Predicting paths. We use the routing policies and algorithmic simulations [22] as described above to compute routes between pairs of ASes using the AS-level topology published by CAIDA [23]. AS-level path prediction between a source and destination is a thorny issue, for example the recent work from Juen, *et al.* [26] shows that the paths predicted by BGP-based path prediction vary significantly from traceroute-based path prediction. However, our BGP-based path prediction toolkit makes use of the state-of-the-art in path inference and AS-relationship inference that have both been extensively validated with empirical measurements by Anwar *et al.* [10] and Giotsas *et al.* [23].

²In practice, this is done using the distance between routers and router IDs. Since we do not incorporate this information in our model we use a randomized tie break which prevents certain ASes from “always winning”.

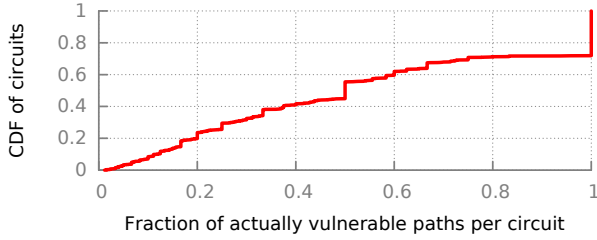


Fig. 3: Fraction of actually vulnerable paths from all possible paths, for each of 20,000 circuits marked as vulnerable by our toolkit.

In particular, Anwar, *et al.* [10] show that 65-85% of measured paths are in the set of paths which satisfy *LP* and *SP*. Thus, we modify the algorithmic simulator to return all paths satisfying *LP* and *SP* simultaneously, instead of using *TB* to produce a unique path. Thus we consider the set of ASes in the set of paths satisfying *LP* and *SP* between *a* and *b* to be the set $\mathcal{P}_{a \rightarrow b}$.

Identifying vulnerable circuits. Let $p_{src \leftrightarrow entry}^i$ be the i^{th} *LP* and *SP* satisfying (forward- or reverse-) path between the source and entry-relay, $p_{exit \leftrightarrow dst}^j$ be the j^{th} such path between the exit and destination, $\mathcal{P}_{src \leftrightarrow entry} = \cup_i \{p_{src \leftrightarrow entry}^i\}$, and $\mathcal{P}_{exit \leftrightarrow dst} = \cup_j \{p_{exit \leftrightarrow dst}^j\}$. We refer to $\mathcal{P}_{a \leftrightarrow b}$ as the path-set between *a* and *b*.

Since it is currently not possible to predict exactly which path from $\mathcal{P} = \mathcal{P}_{src \leftrightarrow entry} \times \mathcal{P}_{exit \leftrightarrow dst}$ will be utilized when using a circuit with entry-relay *entry* and exit-relay *exit*, we label all paths $p \in \mathcal{P}$ as vulnerable *iff* at-least one of the paths in \mathcal{P} is vulnerable (as defined in Eq. 1). That is, once our path prediction toolkit returns the set of ASes that occupy each path-set between the Tor client and a given entry-relay ($\mathcal{P}_{src \leftrightarrow entry}$) and between the exit-relay and destination ($\mathcal{P}_{exit \leftrightarrow dst}$), potential circuits using the corresponding entry- and exit-relay are labeled as vulnerable *iff* there are common or sibling ASes on the (client, entry-relay) and (exit-relay, destination) path-set – i.e., $\{\mathcal{P}_{src \leftrightarrow entry} \cap \mathcal{P}_{exit \leftrightarrow dst}\} \neq \emptyset$. This provides an estimate on the threat posed by network-level attackers.

To understand the tightness of this estimate, we analyzed the fraction of the actually vulnerable paths in each of 20,000 unique “vulnerable” circuits generated by our experiments. Figure 3 shows the result of this analysis. 25% of all circuits had all their paths in \mathcal{P} vulnerable to at-least one network-level attacker and 56% of all circuits had at-least 50% of their paths (in \mathcal{P}) vulnerable to at-least one network-level attacker.

B. Measurement methodology and results

To understand the threat posed by the adversary described in Section II, we performed several experiments. In particular, our goal was to understand the threat faced by the Tor client under various configurations, and in different network and geographic locations.

Experimental setup. In our experiments, we consider the fact that Tor users in different countries face different levels of threats from local ASes. To this end, each experiment

was performed in 10 different countries: Brazil (BR), China (CN), Germany (DE), Spain (ES), France (FR), England (GB), Iran (IR), Italy (IT), Russia (RU), and the United States (US). This list was obtained by considering the intersections of the number of Tor users in each country [42] and the Freedom House rankings for Internet freedom [19]. In order to completely understand the threats faced by Tor users, five experiments were conducted in each country; a summary of each experiment is shown in Table I.

For each experiment, 200 websites were loaded using the Selenium Firefox webdriver [6]. The list of 200 websites comprised of the local Alexa Top 100 sites [1] and 100 sensitive (i.e., likely to be blocked) pages obtained from the Citizen Lab testing list repository [2] for each country.

Each experiment was conducted in one of two settings: Live or Simulation. In the Live setting, the actual client (vanilla Tor or Astoria) being studied was used to load pages from within the respective country using a single VPN as the vantage point. The VPN vantage point only presents a limited picture of the threat faced by all users in the country (since it only considers a single AS as the client location (source AS)), thus we used simulations to augment the Live experiments. Each simulation considered clients located in 100 randomly selected ASes in each country.

For each experiment, logs were maintained to track: (1) the list of available entry- and exit-relays during circuit construction, (2) the actual chosen entry and exit-relay for each circuit constructed by the client, and (3) the list of requests made for each site and the circuit used by the Tor client to serve the request. Data from these logs were fed to our measurement toolkit in order to identify (1) the set of attackers that threaten actually constructed circuits (Live experiments) and (2) the set of attackers that threaten potential circuits – i.e., circuits that could have been constructed given a particular valid combination of available entry- and exit-relays (Simulation experiments).

E1: Measuring vulnerability to network-level attacks. This experiment was conducted using the vanilla Tor client and a modified Tor client using a uniform relay-selection strategy. Both clients used the same VPN in each of the 10 countries to load their corresponding Alexa top 100 and 100 sensitive pages. Three statistics were measured: (1) The number of websites which had the circuits carrying the request for their main page being vulnerable, (2) the number of websites which had any of their circuits being vulnerable, and (3) the total number of vulnerable circuits.

A summary of these results are illustrated in Table II. We see that both clients have similar number of compromisable circuits, however the vanilla Tor client allows 16% more websites to load without having any of their circuits compromised, implying that when a website is loaded with the vanilla Tor client it is either completely safe or has most of its content loaded via a vulnerable circuit. This is due to the fact that unlike the modified Tor client, the vanilla Tor client reuses a small number of circuits for many requests.

We break down our results for the vanilla Tor client by country in Figure 4. The figure shows the percentage of websites that are vulnerable to asymmetric correlation attacks

ID	Question Answered	Vantage Point	Setting	Results
E1	How vulnerable are circuits to asymmetric correlation attacks?	VPN	Live (3 guards)	Figures 4, 14a and 14b
E2	How many <i>attacker-free</i> paths are available to the vanilla Tor client in each country?	100 ASes per country	Simulation (all entry- and exit-relays)	Figures 14 and 6
E3	How much of a threat do colluding sibling ASes pose?	VPN	Live (3 guards)	Figures 7, 14c, and 14d
E4	How much of a threat do state-level attackers pose?	VPN	Live (3 guards)	Figures 8, 14e, and 14f
E5	Do guard settings have a significant effect on the availability of attacker-free paths to the vanilla Tor client?	100 ASes per country	Simulation (20 guard-sets of 1,2, and 3 guards and all exit-relays)	Figure 9

TABLE I: Summary of security experiment settings used for the evaluation of the vanilla Tor client and Astoria. For each country, all experiments used a dataset containing the local Alexa Top 100 and 100 locally sensitive websites (obtained from the Citizen Lab testing repository [2]).

Vulnerable	Vanilla Tor	Uniform Tor
Websites (Main request)	37%	35%
Websites (Any request)	53%	69%
Circuits (All requests)	40%	39%

TABLE II: Summary of threat from asymmetric correlation attacks against the vanilla Tor and uniform relay-selection strategies for 200 websites in 10 countries.

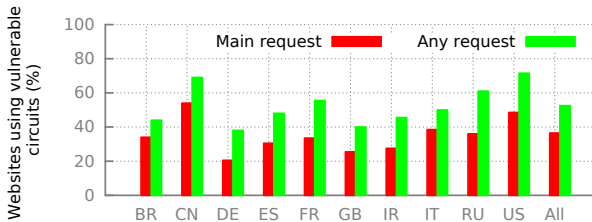


Fig. 4: An estimate of the percentage of websites that have main page requests and any requests serviced by a vulnerable Tor circuit.

on circuits built for serving the request for their main page (GET) and for serving any request. We find that the threat is not uniformly spread. Clients using the vanilla Tor client from our VPN vantage point in three countries: China (CN), Russia (RU), and the United States (US) were found to be most vulnerable. This can be explained by the fact that of our 10 countries, the US, RU, and CN had the most amount of locally hosted content (i.e., content hosted within the country). Of the 200 sites used for each of the countries, 95% (US), 57% (RU), and 47% (CN) made requests to ASes within the country itself – making it more likely for the same AS to be on paths from/to client to/from entry-relay and exit-relay to/from destination.

E2: Measuring fraction of *available attacker-free* paths.

Since the results of our experiments on the live Tor network were highly dependent on the location of the VPN, simulations were required to understand the distribution of threat in other locations within each country. To this end, for each country, 100 ASes were randomly selected as client locations and the targets of each of the requests generated by the 200 sites (sensitive and popular) for each of our 10 countries were used as destinations. The simulation toolkit generated a list of all entry- and exit-relays available to each client for performing the page load (using Tor client consensus data).

Each generated (source, entry, exit, destination) combination was then analyzed for the threat of attackers to understand how many “safe” or “attacker-free” entry-exit pairs were

available. We see in Figure 14 the cumulative distribution function of the fraction of attacker free entry-exit pairs for each source-destination pair. Figure 5a shows this for the five most vulnerable countries in our study, and 5b shows this for the remaining countries.

China (CN) and Iran (IR) stand out as the most interesting cases. First, we see that 8% of all source-destination pairs have less than 10% of their entry-exit options being safe. Next, we also notice that there are no known attackers present on 18% of all source-destination pairs. This appears to indicate that the threat of de-anonymization is non-uniform even within a country, with certain client locations being much safer than others.

In order to understand which set of websites are more vulnerable in each of the countries, in Figure 6 we show the percentage of source-destination pairs having fewer than 5% safe circuit options for each set of websites. We find that in all cases, the Alexa top 100 local websites have fewer safe circuit options. This can be explained by the fact that locally popular websites are likely to be hosted within a regional AS. Additionally, we find that China and Iran have a significant number of their source-destination pairs having fewer than 5% safe circuit options – i.e., over 8% of the source-destination pairs have less than 5% of all their circuit options being safe from network-level correlation attacks.

However, in general, the results of **E1** and **E2** indicate that although in most cases there are many safe entry-exit options available to the Tor client, it often does not select these options – leading to a large number of vulnerable circuits being created.

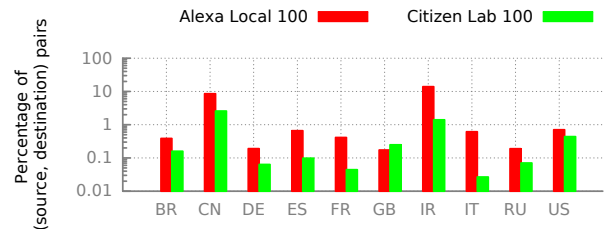


Fig. 6: (Logscale) Percentage of (source, destination) pairs having fewer than 5% attacker-free (entry, exit) options in each country.

E3: Measuring the impact of sibling ASes. In this experiment we consider the possibility that ASes owned by the same organization (referred to as sibling ASes) may collude with each other in order to de-anonymize Tor users via asymmetric

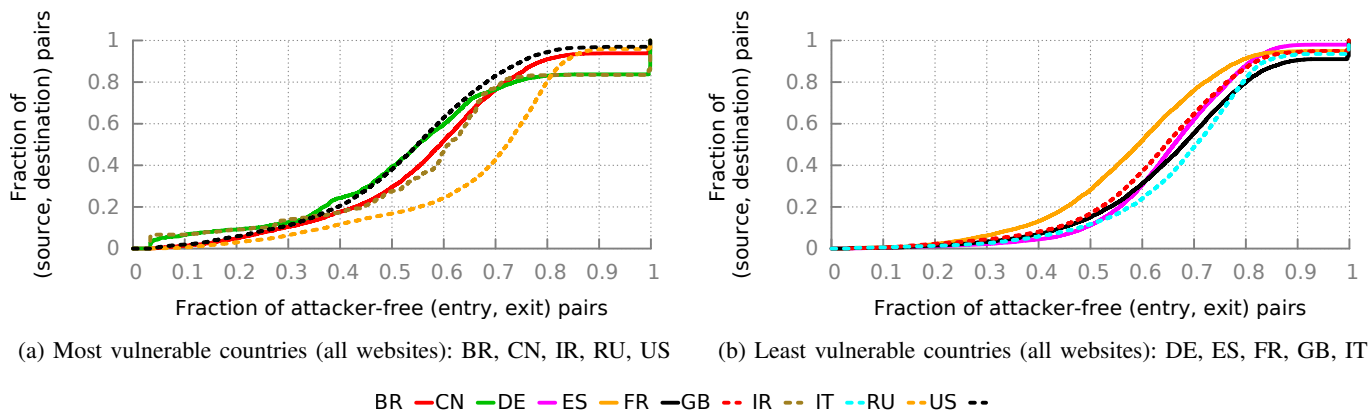


Fig. 5: Distribution of the fraction of attacker-free circuits for 100 source ASes connecting to 200 websites in 10 different countries of interest. More skewed to the right indicates the availability of more safe circuits.

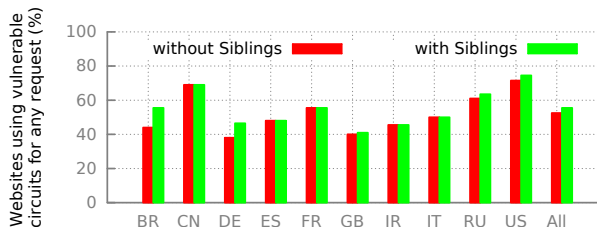


Fig. 7: An estimate of the percentage of websites that have any requests served by a vulnerable Tor circuit when considering siblings.

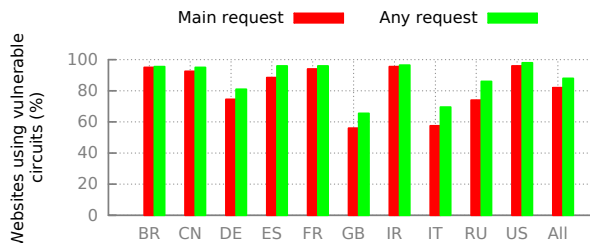


Fig. 8: An estimate of the percentage of websites that have main page requests or any requests served by a vulnerable Tor circuit when considering state-level adversaries.

correlation attacks. We use data gathered by Anwar *et al.* [10] to identify such ASes. The same setup as **E1** was used.

We observe from Figure 7 that the increase in threat from considering sibling ASes is marginal. Over the 10 countries, only 3% additional websites from our list of 200 for each country had some request served by a circuit that was vulnerable to asymmetric attacks by sibling ASes. However, the increase in threat is not uniform. Clients in Brazil and Germany face an 8-10% increase in vulnerable websites. This can be attributed to the large telecom conglomerates operating within the countries – *e.g.*, many paths from our vantage points in Germany and Brazil were vulnerable to correlation attacks due to transiting one of the large number of ASes owned by Telefonica (in Spain) and Durand (in Brazil), respectively.

E4: Measuring the impact of state-level adversaries. In this experiment we consider the threat that Tor clients face from state-level adversaries. We assume that a state-level adversary is able to gain insight into the traffic flowing through all ASes operating within the state. Therefore, we consider a circuit originating from country X to be vulnerable if its path to/from its entry-relay and from/to the exit-relay to the destination contains some AS operating within X. The same setup as **E1** was used for data collection.

The results are broken down per country in Figure 8. Here,

we see that the situation is quite dire with 82% of all (over all 10 countries) websites having their main page served by a vulnerable circuit. In particular, clients in Brazil, China, France, Iran, and the United States face the biggest threat from state-level attacks with over 95% of their main page requests being vulnerable to state-level attackers.

E5: Measuring the effect of guards. In this experiment we consider the effect of the number of guards on the vulnerability of Tor clients to network-level asymmetric correlation attacks. For each of our 10 countries, 100 ASes were randomly selected as client locations and the targets of all the requests generated by the 200 websites in our earlier experiments were used as the destinations. The simulation toolkit generated 60 unique guard-sets (20 each for 3 guards, 2 guards, and 1 guard) in an identical manner to the vanilla Tor client, and a list of all exit-relays available to each client for performing the page load (using Tor consensus data). Each (source, entry, exit, destination) combination was checked for the presence of our adversary.

Figure 9 illustrates the effect that reducing the size of the guard-set has on the fraction of network-level attacker-free-paths available to the Tor client.

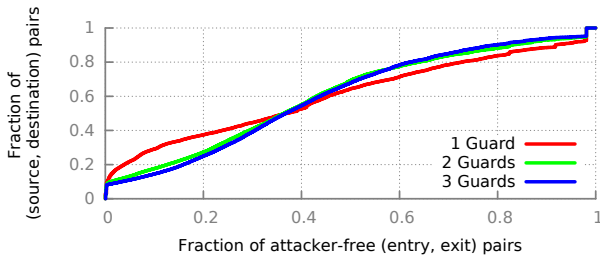


Fig. 9: Distribution of the fraction of attacker-free (entry, exit) pairs for vanilla Tor with 3, 2, and 1 guard(s).

While it is known that a smaller number of guards provides better security against relay-level attackers in the long-term [14], we see from the results of this experiment that the effect is the opposite against network-level adversaries – i.e., as the size of the guard-set decreases, Tor is more likely to select a circuit vulnerable to network-level asymmetric correlation attacks due to the reduced number of available safe paths. In particular, when only 1 guard is used, over 15% of the (source, destination) pairs in our experiment had no safe-options, whereas the difference in security provided by two or three guards was marginal. This experiment demonstrates one of the conflicts between Tor clients geared for defending against relay-level attackers and those geared for defending against network-level attackers.

IV. ASTORIA: AN AS- AND CAPACITY-AWARE TOR CLIENT

Motivated by the observation that vanilla Tor very often selects entry-exit pairs that may be subject to asymmetric correlation attacks, we seek to design a relay selection algorithm to mitigate the opportunities for such attackers. We design our relay selection system, Astoria, based on the idea of stochastic relay selection. This works by having the Tor client generate a probability distribution that minimizes the chance of attack over all possible entry- and exit- relay selection choices, and selecting an entry- and exit-relay based on this distribution. The advantage of stochastic selection is that even if the client has no safe options, relay-selection can be engineered to minimize the amount of information gained by the adversary over some period of time (as we show below). Further, it allows clients to select relays in a way such that no set of relays in the Tor eco-system is overloaded, even if every client uses the same relay-selection strategy.

A. Astoria goals

Astoria is constructed with several security and performance goals in mind:

- *Deal with asymmetric attackers.* Astoria avoids constructing circuits involving common ASes on the forward- or reverse-paths between the client to the entry-relay and the exit-relay and the destination.
- *Deal with the possibility of colluding attackers.* Astoria considers the threat of ASes that may collude to de-anonymize Tor users. Astoria can be configured to

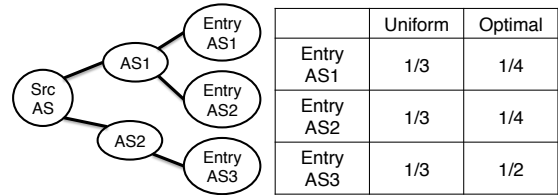


Fig. 10: Example of optimizing relay selection. Simplified to unidirectional paths and only entry-relay selection.

build circuits that do not contain known to be colluding ASes on the forward- or reverse-path between the client and entry-relay and exit-relay and destination. This mitigates the threat from sibling ASes and state-level attackers.

- *Consider the worst case possibility.* Astoria uses a probabilistic relay selection algorithm that ensures, even in the worst-case (where there are no safe paths to and from the entry- and exit-relay), that the ability of a single AS (or, family of ASes) to de-anonymize a large number of circuits is minimized.
- *Minimize performance impact.* It is clear that any AS-aware client will lose its ability to perform many optimizations such as pre-constructing circuits. Our goal is to minimize the effect of the above considerations on the performance of the Tor client.
- *Be a good network citizen.* Astoria takes into account the capacities of all relays available in the Tor eco-system and performs selection in a way that no single set of relays are overloaded, even when all clients in the network use the same relay-selection strategy.

B. Minimizing information gained by the adversary

While there often are cases when there is a relay selection that will completely eliminate the risk of our adversary, we develop our relay selection to be robust, even if this is not the case. Further, with attacks implemented using BGP hijacking and interception the number of unsafe paths may be higher than what we observe in our analysis (we discuss this more in Section VI).

To minimize the risk of correlation attacks, we define a linear program which generates a probability for each relay selection with the objective to minimize the maximum probability of a circuit encountering the attacker. Recall that in our adversary model, we consider a long-lived adversary and that minimizing the probability of an attacker may also be seen as minimizing the number of circuits the adversary is able to observe over a long period of time and numerous circuit construction cycles.

Figure 10 shows an example of relay selection to give intuition about how the LP minimizes the risk from the attacker. In this example, we consider unidirectional paths and only entry-relay selection for clarity. In the figure, if the source were to choose uniformly at random across the three entry-relays, there is a 2/3 chance that AS1 will be able to observe traffic and only a 1/3 chance that AS2 will. In this case, the

optimal selection is intuitive, that the source should choose entry-relays 1 and 2 with probability 1/4 each and entry-relay 3 with probability 1/2. This lowers the probability that AS1 can observe a circuit from 2/3 to 1/2. This probability of the most likely adversary is the quantity that our LP minimizes.

We use the following notation:

- Let $ADV_{i,j}$ be the set of attackers on the circuit using entry-relay i and exit-relay j to destination $dest$ – i.e., $\forall A \in ADV_{i,j} : A \in \{p_{src \leftrightarrow entry_i} \cap p_{exit_j \leftrightarrow dest}\}$.
- Let $X_{i,j,A}$ be an indicator random variable for attacker A on the circuit using entry-relay i and exit-relay j – i.e., $X_{i,j,A} = 1 \iff A \in ADV_{i,j}$, and 0 otherwise.
- Let $P_{i,j}$ be the probability that a client builds a circuit using entry-relay $entry_i$ and exit-relay $exit_j$.

The following linear program is used to minimize the probability of the most likely attacker (i.e., the number of circuits visible to the attacker).

$$\begin{aligned}
& \text{minimize } z \\
& \text{subject to } z \geq \sum_{i,j} (P_{i,j} X_{i,j,A}) \quad \forall A \in ADV_{i,j} \\
& P_{i,j} \in [0, 1], \quad \forall i, \forall j; \quad \sum_{i,j} P_{i,j} = 1
\end{aligned} \tag{2}$$

Essentially, given information about the presence of attackers (network-level or state-level) for each $p_{source \leftrightarrow i}$ and $p_{j \leftrightarrow dest}$ path, the linear program seeks to find the probability distribution ($P_{i,j}$) over available choices of entry- and exit-relays, for which the expected number of circuits visible to each attacker is minimized. Entry- and exit-relays are chosen according to this distribution (defined as D_{lp}) during circuit construction.

C. Security is not enough

While our LP produces a relay selection distribution that minimizes the probability of success across all adversaries, it does not take into account the resources available at the selected relays. Given that Tor is a system run using community resources contributed by volunteers, load balancing users across these resources is important to ensure that they are used efficiently and no single relay or set of relays become overloaded. Figure 12 shows a snapshot of the distribution of relay capacities available during the period of this study, for all relays in the Tor system and the relays selected by a hypothetical perfect load-balancing Tor client – i.e., one where each relay serves exactly the amount of traffic that it can handle (assuming identically sized requests). Here, we see that over 80% of all Tor traffic should be routed through $\approx 35\%$ of all the relays in the Tor network for every relay to be operating within its advertised capacity.

In order to achieve load-balancing, we augment our relay-selection algorithm with information about relay capacities from the latest Tor consensus during circuit construction. This is done as follows:

When there are safe entry and exit combinations: In this case, we select a safe combination according to the distribution

of relay capacities. For example, given a set of safe entry- and exit-relay combinations $E = \{(en_1, ex_1) \dots (en_n, ex_n)\}$ and the distribution of their advertised capacities $D_{bw} = \{en_1, \dots, en_n, ex_1, \dots, ex_n\}$, we select a combination (en_i, ex_i) with probability $P_i = \frac{D(en_i) \times D(ex_i)}{\sum_{j=1}^n D(en_j) \times D(ex_j)}$.

This ensures that no single (entry- or exit-) relay is selected with probability higher than the ratio of its advertised capacity and the total advertised capacity of all safe (entry- or exit-) relays (just as is done by the vanilla Tor client).

When there are no safe entry- and exit-relay combinations: In this case, in order to correctly minimize the amount of information gained by the adversary, we strictly obey the probability distribution output by our linear program described in the previous section. No attempt is made to balance loads according to relay capacities. It is important to note that this is a fairly infrequent case (as shown in experiment E2 in Section III).

D. Implementing Astoria

The measurement toolkit described in Section III was integrated with a modified Tor client, as follows.

Integrating our path measurement toolkit with the Tor client. For standard measurement purposes, the toolkit simply takes a source and destination address and returns the set of ASes on the forward and reverse-path between the two.

However, in the context of integration with the Astoria client, it must predict paths to and from each of the entry-relays for the client’s AS, and paths from all exit-relays toward the destination AS (Figure 2). This results in $|En| + |Ex| + 2$ routing-tree computations where $|En|$ and $|Ex|$ are the number of entry and exit relays, respectively. In order to mitigate the risk of correlation attacks, by default, Tor restricts the number of entry-relays available to each Tor client to three (called guards [14]), and there are typically of the order of 1,000 exit-relays available to a client during circuit construction – resulting in the order of 1,000 routing-tree computations.

Fortunately, since the source AS and entry-relay ASes are relatively stable, these paths can be precomputed for later use by the client. (We observe the benefit of this in Section V.) However, performing relay selection on a per-destination basis means that pre-building circuits, as is done by the current implementation of Tor, is no longer feasible.

AS-aware on demand circuits. First, the Tor client was modified to perform offline IP to ASN mapping using a database [38] for every incoming request. Note that since the entire database (9 MB) is downloaded, the client does not reveal its intended destination to any lookup services.

Next, modifications were made to the way requests were allocated to circuits. The vanilla Tor client performs pre-emptive circuit construction in order to serve requests as they arrive (increasing performance significantly). This is unfortunately infeasible for a AS-aware client where relay-selection is a function of the destination. Although one may consider pre-constructing AS-aware circuits for a set of popular destination ASes, the performance benefit is marginal, at best. This is mainly due to the large number of third party requests for less

popular destination ASes embedded in popular Web pages. Astoria, therefore, only performs on demand circuit construction. For each incoming request, Astoria first checks if there are existing circuits serving the same destination AS. The request is attached to the most suitable such circuit if it exists.

Circuit construction. Astoria creates a new circuit if and only if a request arrives for a destination with no currently usable circuits. In such cases, the client and destination ASNs are passed to the circuit construction and relay selection algorithms. Circuit construction is performed as follows:

- First, a list of entry- and exit-relays meeting the requirements set by the request were obtained. If the Tor client is configured to utilize only guards as entry-relays, the list of guards is obtained. Next, in order to perform load-balancing, information from the most recent Tor consensus is obtained to generate the relay capacity distribution D_{bw} for each entry- and exit-relay combination.
- The Astoria client performs lookups to the offline IP-ASN database to perform mapping between entry- and exit-relay IP address and AS numbers. These, along with the client and destination AS numbers are then passed to our AS-path prediction and attacker measurement toolkit (Section III).
- The toolkit returns the list of ASes on each forward- and reverse-path between the client and every potential entry-relay and the destination and every potential exit-relay. In order to improve performance, paths are cached for frequently queried destinations. Precomputation or caching of paths between the client and the high-uptime entry-relays and destinations and high-uptime exit-relays also help improve performance.
- The returned paths are checked for the presence of common ASes in the entry and exit AS path sets. If there are paths without an attacker, the linear program need not be invoked. Instead, Astoria selects a safe entry- and exit-relay combination according to the generated D_{bw} probability distribution (described in Section IV-C). We see the impact of this load-balancing technique in Section V.
- If there are no attacker-free relay combinations, the linear program is invoked in order to select an entry- and exit-relay combination according to the distribution D_{lp} that minimizes the probability of the most likely attacker (described in Section IV-B).
- Finally, once the entry- and exit-relays are selected according to one of the D_{bw} or D_{lp} distributions, the circuit is constructed. The remainder of the circuit construction process remains unchanged from the vanilla Tor client.

V. ASTORIA EVALUATION

We evaluate Astoria along multiple axes. First, we consider the performance of Astoria by measuring the time required to load webpages and its ability to be a good Tor citizen by selecting bandwidth-rich relays. Second, we evaluate the security provided by Astoria. We show that Astoria constructed

circuits are a good defense against the adversary described in Section II. Finally, we evaluate the threat from attacks by relay-level adversaries.

A. Evaluation methodology

Similar to our experiments in Section III, we consider the performance and security of clients in 10 different countries – Brazil (BR), China (CN), Germany (DE), Spain (ES), France (FR), England (GB), Iran (IR), Italy (IT), Russia (RU), and the United States (US). The same 200 webpages as before were used for page-loads within each country.

In order to understand the performance of Astoria and for comparison with the vanilla Tor client, three metrics were computed: (1) page-load times³, (2) distribution of selected relay bandwidths, and (3) overhead of path prediction. For each of these experiments we considered the same experimental settings as the vanilla Tor client in experiment **E1**. Logs were recorded to extract advertised capacities of all available relays and all relays selected by the Astoria and vanilla Tor clients, and time required for AS path computation by the Astoria client.

In order to assess the security of Astoria and for comparison with the vanilla Tor client, experiments to measure security against network-level (experiment **E1**), colluding network-level (experiment **E3**), and state-level (experiment **E4**) asymmetric correlation attackers were repeated using the Astoria client for page-loads in the same setting (including using the same guard-set in each country) as the vanilla Tor client (Section III). For each experiment, three statistics were computed: (1) the fraction of websites whose main page requests were served by vulnerable circuits, (2) the fraction of websites that any request that was served by a vulnerable circuit, and (3) the total fraction of vulnerable circuits.

B. Performance evaluation

In this section, we evaluate the performance of Astoria using three metrics: (1) page-load times, (2) distribution of selected relay bandwidths, and (3) overhead of path prediction.

Page load times. Figure 11 shows the distribution of page-load times when using the vanilla Tor client, a modified Tor client with a uniform relay-selection strategy, and the Astoria client. We see that the median page-load time with the vanilla Tor client is only **5.9** sec, while the median page-load time for the Astoria and uniform Tor client are **8.3** sec and **15.6** sec, respectively. Although this drop in performance from the vanilla Tor client to Astoria is significant, it can be argued there are two main causes for this, both of which are unavoidable to any AS-aware Tor client: (1) It is no longer possible to pre-construct and re-use circuits to the same degree as the vanilla Tor client, and (2) There is a non-negligible amount of time spent for computing paths and checking for the presence of attackers on these paths.

Load balancing. Astoria aims to balance load from clients across all relays in the Tor network so that no single set of relays are overloaded. Figure 12 demonstrates the closeness

³The Selenium `driver.get()` method was used to detect the end of page-loads.

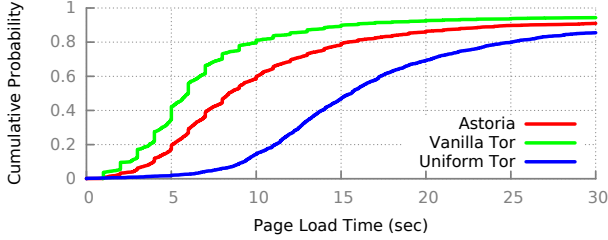


Fig. 11: CDF of page load times (including circuit creation times) for a uniform Tor, vanilla Tor, and Astoria client over 200 websites in all 10 countries.

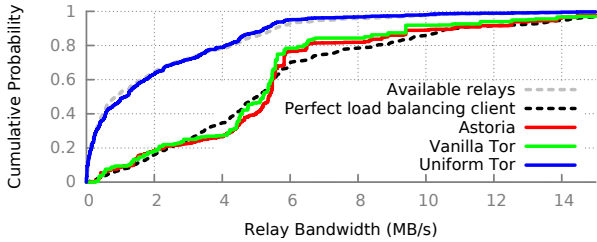


Fig. 12: Distribution of bandwidths of relays selected by vanilla Tor, uniform Tor, Astoria, and the perfect load balancing client.

of the load-balancing of the Astoria client with the vanilla Tor client and the perfect load-balancing client. We see that in spite of performing AS-aware relay-selection, Astoria is able to perform load-balancing at least as well as the vanilla Tor client, with neither of them achieving a perfect distribution.

The results of this experiment allow us to confirm our hypothesis that the reduction in performance from the vanilla Tor client to Astoria is indeed because of our inability to preconstruct circuits and delays due to path computation, and not due to poor relay-selection.

Overhead of path prediction. Figure 13 shows the CDF of the total amount of time spent on computing AS paths, for each site. We see that for about 50% of all sites (200 sites in each of 10 countries), the time spent on path computation is negligible. This is due to the high frequency of repeated occurrences of destination ASes in our 200 sites – resulting in the AS path for each exit-relay to that destination already being in the toolkit’s cache. In 60% of the cases where responses were not cached (and 86% of the cases, overall), computing AS paths required under 4 seconds.

C. Security against network-level attackers

In this section, Astoria is evaluated and compared with the vanilla Tor client by measuring its success in defending against various attackers performing asymmetric correlation attacks. A summary of all results are provided in Table III.

E1: Measuring vulnerability to network-level attacks. In this experiment, we compare the security provided by the Astoria client with the vanilla Tor client, against network-level adversaries. The threat from such adversaries is significantly

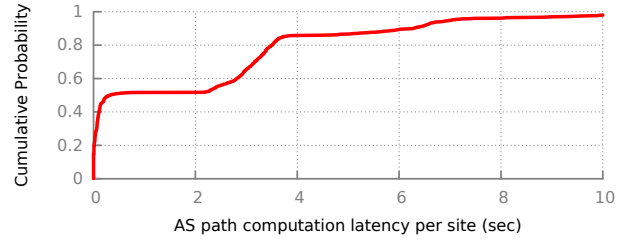


Fig. 13: CDF of time spent on AS path computation per site.

reduced from up to 40% of all circuits being vulnerable to 3%, with the Astoria client. Figures 14a and 14b breaks down the results of this experiment by country. We see that Astoria completely removes the threat of network-level attackers on circuits carrying the main page request in clients from Brazil, France, and Iran, while bringing the risk down to under 5% in six other countries.

E3: Measuring the impact of sibling ASes. We find that siblings have little impact on the security provided by Astoria. Over all circuits constructed by Astoria, the addition of colluding sibling ASes resulted in less than a 3% increase in number of vulnerable circuits, with the only significant increase being in Germany (DE). This is illustrated in Figures 14c and 14d. This large increase in number of vulnerable circuits indicates that if sibling ASes in Germany were to collude, Astoria (given the VPN client location and selected entry-guards) is often left with no safe entry- and exit- relay options for circuit construction. It is important to note that although there are a significant number of vulnerable circuits created by Astoria, these circuits are constructed using our linear program (Eq. 2) which minimizes the number of circuits visible to each attacker.

E4: Measuring the impact of state-level adversaries. Astoria performs reasonably well even against state-level adversaries by reducing the fraction of potentially vulnerable circuits from 85% (vanilla Tor) to 25%, over all countries. The per country breakdown is illustrated in Figures 14e and 14f. The results show a steep decrease in the ratio of vulnerable websites for all countries except the United States (US). This is due to the large presence of American ASes on paths to and from our US VPN vantage point and the entry-guards and any Tor exit-relay and our US destinations.

Defending against active network-level attacks. Astoria focuses on adversaries who may lie on asymmetric network paths between the client and entry; and exit and destination, respectively. However, Sun *et al.* [39] highlight attacks based, not only on static path properties, but also dynamics of BGP (e.g., hijacks, routing instability). Taking this sort of attack into account is challenging as it requires realtime access to interdomain routing data and intelligent analysis to identify incidents that may impact the safety of the client’s path. In the future, we plan to integrate subscriptions to BGP hijack data sources (e.g., Argus [36], or ongoing efforts at building a real-time interception detector [12]) into Astoria to allow it to operate on dynamic BGP paths.

Client	Network-level (E1)			Colluding network-level (E3)			State-level (E4)		
	Websites (Main)	Websites (Any)	Circuits (All)	Websites (Main)	Websites (Any)	Circuits (All)	Websites (Main)	Websites (Any)	Circuits (All)
Astoria	3%	8%	2%	6%	13%	5%	27%	34%	25%
Vanilla Tor	37%	53%	40%	40%	56%	42%	82%	88%	85%

TABLE III: Astoria vs. vanilla Tor: An estimate of the threat faced from various attackers.

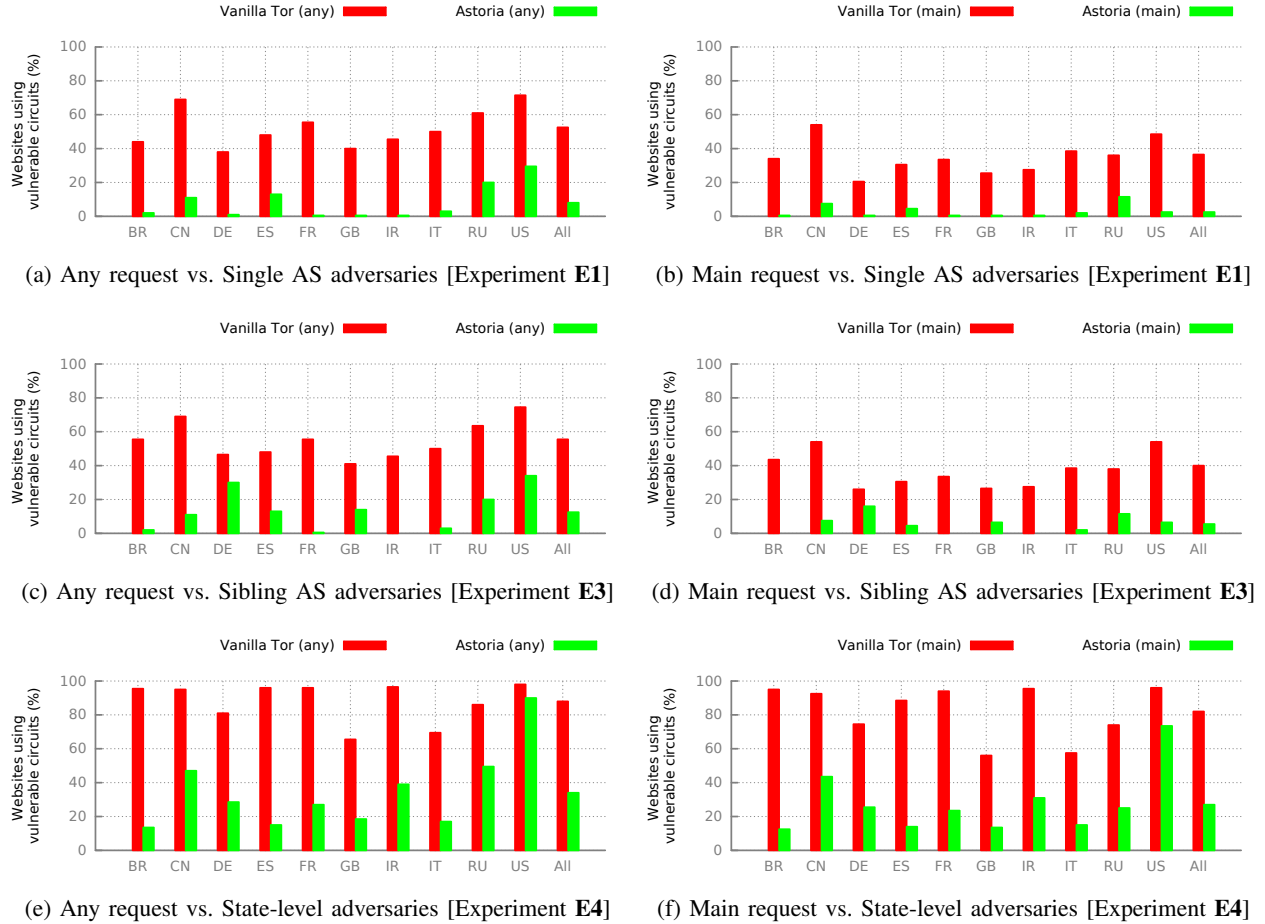


Fig. 14: Astoria vs. vanilla Tor: Percentage of websites using vulnerable circuits for their main request or any request, against various adversaries.

D. Security against relay-level attackers

In order to defend against relay-level attackers, Astoria inherits the concept of entry-guards from the vanilla Tor client and also ensures that no two relays from the same family are placed on the same circuit. However, due to its AS-awareness, Astoria (and any AS-aware client that constructs circuits which are a function of the destination AS) currently is vulnerable to two relay-level attacks: (1) it is possible for a middle-relay in an Astoria constructed circuit to narrow down the set of possible (source, destination) AS pairs that are at either end of the circuit (based on the selected entry- and exit-relays), and (2) when Astoria is used from regions with no safe (entry, exit) relay options, it is possible for a relay-level attacker to force Astoria to create circuits that can be de-anonymized by it. Below, we discuss these attacks, their impact, and how to mitigate them.

Measuring the threat posed by middle-relays. As seen in Table III, in a majority of all cases, Astoria is able to find a safe pair of entry- and exit-relays to use for its circuits. As a result, an adversarial middle-relay working under the assumption that Astoria always constructs safe circuits, will be able to narrow down the set of possible source- and destination-ASes by simply observing the entry- and exit-relays in the circuit. Below, using the results of experiment E2 and statistical inference techniques, we show that the threat from such adversarial relays is negligible.

First, given our random sample of 100 source ASes for each country (and fixed set of destinations) we infer the mean number of (source, destination) pairs with greater than **50%** safe entry- and exit-relay pair options for the entire population of source ASes in each country (with the same fixed destinations). Then, we find a lower-bound estimate on the expected number of (source, destination) AS pairs that have

each (entry, exit) pair as a safe option – i.e., a lower-bound on the number of (source, destination) pairs that can be linked to the circuit by a middle-relay in a single observation. Finally, we show that given the current distribution of Tor relays, the probability of narrowing down this set of sources to a single (source, destination) pair is negligible.

Inferring the mean number of (source, destination) pairs with greater than 50% safe options. Recall that in experiment **E2**, 100 source ASes were selected at random from the set of all ASes in each country. The experiment considers the destination ASes generated by the loading of 200 non-random destinations. Let the set of sampled source ASes be denoted by \bar{X} and the set of destination ASes be denoted by D . From the results of the experiment, we extract the mean fraction of $(\bar{x} \in \bar{X}, d \in D)$ pairs which have more than 50% safe entry- and exit-relay options (denoted by $\mu_{\bar{X},D}$). Let X denote the set of all ASes within each country. Now, using the central limit theorem and the sampling distribution of the sample means [34], we infer the 99% confidence-interval for the mean fraction of $(x \in X, d \in D)$ pairs which have more than 50% safe entry- and exit-relay options (denoted by $\mu_{X,D}$).

Estimating a lower-bound on linkable sources. We take an extremely conservative approach to derive this lower-bound. First, we use the lower value of $\mu_{X,D}$ from our 99% confidence interval. Further, we assume that $\mu_{X,D}$ fraction of our $(x \in X, d \in D)$ pairs have only exactly 50% safe entry- and exit-relay options (although $\mu_{X,D}$ denotes the fraction of $(x \in X, d \in D)$ pairs with greater than 50% safe options). Finally, we assume that the remaining $1 - \mu_{X,D}$ fraction of $(x \in X, d \in D)$ pairs have no safe options. Given these assumptions, we can compute the lower-bound on the expected number of $(x \in X, d \in D)$ pairs which have each (entry, exit) pair as a safe option (denoted by $E[S_{en,ex}]$) as: $E[S_{en,ex}] = \frac{\text{Total safe circuits}}{\text{Total (entry, exit) pairs}} = .50 \times \mu_{X,D} \times |X| \times |D|$.

$E[S_{en,ex}]$ is a lower-bound on the expected number of linkable source and destination pairs for each observation of an entry- and exit-relay (under the conservative assumption that an adversarial middle-relay knows the country in which the client is located and the set of all possible destinations D that any client may connect to).

Estimating the probability of complete de-anonymization. Given that $E[S_{en,ex}]$ is the number of $(x \in X, d \in D)$ pairs that are linkable to a single observation of an (entry, exit) pair and assuming a constant rate of reduction in linkable pairs (given by $\frac{E[S_{en,ex}]}{|X| \times |D|}$), the number of circuits that need to be observed by the adversarial middle-relay to narrow down the number of $(x \in X, d \in D)$ pairs to 1 – i.e., to completely de-anonymize the source and destination – is $n = \frac{-\log(|X| \times |D|)}{\log(E[S_{en,ex}]) - \log(|X| \times |D|)}$ (since $(\frac{E[S_{en,ex}]}{|X| \times |D|})^n = \frac{1}{|X| \times |D|}$).

Since Astoria (1) constructs new circuits only if there are no existing circuits that serve the same destination AS, and (2) selects middle-relays for each new circuit according to the bandwidth distribution of relays, we obtain the expected upper-bound of the probability of a middle-relay being able to observe n circuits between the same source and destination ASes (with different entry- and exit-relays). Table IV shows that this probability (denoted by P_n) is negligible even for the Tor relay with the current highest advertised bandwidth where the probability of selection as the middle-relay is **.007**.

	$ X $	$ D $	$\mu_{\bar{X},D}$	99%CI $\mu_{X,D}$	E [S]	n	P_n
BR	3,515	165	.40	(.39, .41)	114,797	8.1	5.7×10^{-18}
CN	1,227	131	.44	(.43, .46)	35,216	7.8	8.2×10^{-16}
DE	2,022	190	.33	(.33, .34)	63,409	7.1	8.2×10^{-16}
ES	703	181	.40	(.39, .41)	25,295	7.2	8.2×10^{-16}
FR	1,251	187	.32	(.31, .33)	36,448	6.6	1.1×10^{-13}
GB	2,372	187	.35	(.34, .36)	76,473	7.3	8.2×10^{-16}
IR	470	133	.39	(.38, .40)	11,878	6.6	1.1×10^{-13}
IT	932	201	.29	(.28, .30)	26,800	6.2	1.1×10^{-13}
RU	5,868	178	.27	(.26, .28)	140,201	6.9	1.1×10^{-13}
US	23,588	188	.45	(.44, .46)	977,768	10.1	2.8×10^{-22}

TABLE IV: Results from statistical analysis of the expected upper-bound of the threat posed by adversarial middle-relays on Astoria (using data obtained from our simulation experiment (**E2**)).

Defending against attacks due to predictable relay-selection when there are no safe options. In certain client locations (e.g., some ASes in China and Iran), there are no safe entry- and exit-relay selections for some destinations, regardless of the guards used by the client. In these cases, a relay-level adversary may place entry- and exit-relays in ASes that provide a safe-path for Astoria clients attempting to connect to specific target destinations. This manipulates Astoria into using the adversarial (entry, exit) pair on all circuits connecting the client to the target destination – allowing trivial de-anonymization of the user.

Astoria can defend against such attacks by selecting from safe (entry, exit) pairs only when a minimum threshold of available safe (entry, exit) pairs is met. In cases where the threshold is not met, Astoria may discard the few remaining safe pairs and choose entry- and exit-relays according to the distribution produced by its linear program (Eq. 2), which minimizes the amount of information gained by the network-level adversary. This however, enables correlation attacks by selected network-level attackers. Since it is not yet clear if network-level adversaries pose a larger threat than relay-level adversaries. Therefore, determining this threshold is a non-trivial open research problem.

VI. DISCUSSION

In this section, we compare the Astoria Tor client with the hypothetical perfect Tor client and discuss how Astoria can be augmented and improved with recent and ongoing developments from the network measurement community.

A. Comparing Astoria and the perfect Tor client

Here we point out some of the shortcomings of Astoria when compared to the perfect Tor client. We find that many of these apply to any AS-aware client. The perfect Tor client is able to simultaneously achieve three conflicting goals:

Defend against network-level attackers. The perfect Tor client is able to prevent compromise from network-level at-

tackers. In particular, the client constructs circuits that are safe from traffic correlation attacks.

While such adversaries are largely ignored by the vanilla Tor client, Astoria successfully deals with them by utilizing efficient path-prediction tools to explicitly avoid relays that enable correlation attacks. However, Astoria does not currently deal with attacks from active network-level adversaries that are able to exploit BGP dynamics. In addition, Astoria is unable to exactly predict the paths that will be utilized to communicate with each Tor relay, and therefore only makes estimates (which are validated to be reasonably tight estimates).

Defend against relay-level attackers. Since the Tor network is volunteer driven, it is critical for the perfect Tor client to be able to defend against passive and active attackers that are able to control a fraction of all relays within the network. This primarily involves (1) constructing circuits so that the probability of an adversarial pair of relays occupying the entry- and exit-hop of the circuit is low, and (2) ensuring that no single relay should be able to conclusively link the source and destination of the circuits it is on.

While the vanilla Tor client is able to successfully mitigate threats from many types of relay-level attacks, we find that this is challenging for AS-aware clients such as Astoria. First, while the concept of entry-guards mitigates many threats from relay-level attackers, it has a negative influence on the number of safe circuits that can be built by AS-aware clients. Second, AS-aware circuits inherently leak some information about the source and destination of the circuit. Our analysis in Section V-D shows that in the average-case, Astoria circuits are safe from de-anonymization due to these leaks.

Maintain performance and load-balancing. The perfect Tor client must also perform load-balancing to ensure that no single set of relays in the network are overloaded, while providing reasonable performance for all its users.

In Section V we demonstrated that Astoria performs load-balancing in an identical manner to the vanilla Tor client and page-loads are only slightly slower in most cases. There are two main reasons for Astoria’s increased page-load times: (1) Path prediction is expensive, and (2) Astoria loses the ability to pre-emptively construct circuits. While (1) is unavoidable, there are interesting future research questions regarding (2) – *e.g.*, can smart caching and pre-emptive/predictive circuit construction for a set of popular/predicted destinations result in significant performance gains?

B. Improving path-prediction accuracy

Measuring the potential threat of correlation attacks is made challenging by the fact that it requires measuring both forward and reverse network paths between the client and entry, and exit and destination, respectively. Thus, we opt to leverage an up-to-date map of the Internet’s topology, augmented with inferred business relationships between networks and a model of routing policies to infer network paths. Modeling of interdomain routing is a thorny issue and we take care to avoid well known pitfalls including complex business relationships (*e.g.*, ASes that act as a customer in one geographic region, and a peer in others) and sibling ASes (*ie.*, multiple ASes which correspond to a single organization). The

issue of siblings ASes is particularly relevant in our context, as multiple ASes controlled by a single organization may share information to perform a correlation attack. Despite all this, accurate path prediction remains an open challenge. In a related study, we validate the accuracy of this approach and find that measured paths follow this model 65-85% of the time [10]. As a result, the numbers we observe should be taken as an estimate of the threat.

We note that novel path measurement tools are on the horizon (*e.g.*, Sibyl [17]) that take into account richer vantage point sets than prior work (*e.g.*, PlanetLab used by iPlane [28] vs. RIPE Atlas [35] used by Sibyl). An interesting future direction is determining how such measurement planes can be integrated into a Tor client (*e.g.*, to operate in an offline mode or via a secured querying interface).

VII. CONCLUSIONS

We have leveraged highly-optimized algorithmic simulations of interdomain routing on empirically-derived AS-level topologies to quantify the potential for correlation attacks where an adversary can leverage asymmetric Internet routing and collude with others within the same organization. Our results show that a significant number of Tor circuits are vulnerable to AS- and state-level attackers.

To mitigate the threat from such attackers, we developed Astoria—an AS-aware Tor client. Beyond providing a high-level of security against these attacks, Astoria also has performance that is within a reasonable distance from the current Tor client. Also, unlike other AS-aware Tor clients, Astoria also considers how circuits should be built in the worst case, *i.e.*, when there are no safe relays available to the client. Further, Astoria is a good network citizen and is designed to ensure that the all circuits created by it are load-balanced across the volunteer-driven Tor network.

Our work highlights the importance of applying current models and data from network measurements to inform relay selection so as to protect against timing attacks. Astoria also opens multiple avenues for future work such as integrating real-time hijack and interception detection systems (to fully counter RAPTOR [39] attacks) and understanding how new measurement services can be leveraged by a Tor client without defeating anonymity.

Source code: The source code of the Astoria client is available under the CRAPL ⁴ license at <http://nrg.cs.stonybrook.edu/astoria-as-aware-relay-selection-for-tor/>.

ACKNOWLEDGMENTS

We would like to thank Ruwaifa Anwar, Haseeb Niaz, and Abbas Razaghpanah for their help with integrating sibling detection algorithms into our measurement toolkit.

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1350720, a Google Faculty Research Award, ISF grant 420/12, Israel Ministry of Science Grant 3-9772, Marie Curie Career Integration Grant, Israeli Center for Research Excellence in Algorithms

⁴<http://matt.might.net/articles/crapl/>

(I-CORE), and an Open Technology Fund Emerging Technology Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, Google, the Israel Ministry of Science, or the Open Technology Fund.

REFERENCES

- [1] "Alexa top sites," <http://www.alexa.com/>.
- [2] "Collection of censorship blockpages as collected by various sources," <https://github.com/citizenlab/blockpages>.
- [3] "How the nsa attacks tor/firefox users with quantum and foxacid," https://www.schneier.com/blog/archives/2013/10/how_the_nsa_att.html.
- [4] "The lifecycle of a new relay – the tor blog," <https://blog.torproject.org/blog/lifecycle-of-a-new-relay>.
- [5] "Nsa stores metadata of millions of web users for up to a year, secret files show," <http://www.theguardian.com/world/2013/sep/30/nsa-americans-metadata-year-documents>.
- [6] "Selenium web browser automation," <http://www.seleniumhq.org/>.
- [7] "'Tor Stinks' presentation," <http://www.theguardian.com/world/interactive/2013/oct/04/tor-stinks-nsa-presentation-document>.
- [8] "Torspec – tor's protocol specifications," <https://gitweb.torproject.org/torspec.git/tree/path-spec.txt>.
- [9] M. Akhoondi, C. Yu, and H. V. Madhyastha, "Lastor: A low-latency as-aware tor client," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 476–490.
- [10] R. Anwar, H. Niaz, D. Choffnes, I. Cunha, P. Gill, and E. Katz-Bassett, "Investigating interdomain routing policies in the wild," in *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, ser. IMC '15. New York, NY, USA: ACM, 2015, pp. 71–77.
- [11] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, "Denial of service or denial of security?" in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 92–102.
- [12] CAIDA, "HIJACKS: Detecting and Characterizing Internet Traffic Interception based on BGP Hijacking."
- [13] N. Danner, S. DeFabbia-Kane, D. Krizanc, and M. Liberatore, "Effectiveness and detection of denial of service attacks in Tor," *Transactions on Information and System Security*, vol. 15, no. 3, pp. 11:1–11:25, 2012.
- [14] R. Dingleline, "Improving tor's anonymity by changing entry guard parameters," *The Tor Blog*.
- [15] R. Dingleline, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, August 2004.
- [16] M. Edman and P. Syverson, "As-awareness in tor path selection," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 380–389.
- [17] Ethan Katz-Bassett and Pietro Marchetta and Matt Calder and Yi-Ching Chiu and Italo Cunha and Harsha Madhyastha and Vasileios Giotsas, "Sibyl: A Practical Internet Route Oracle."
- [18] N. Feamster and R. Dingleline, "Location diversity in anonymity networks," in *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, ser. WPES '04. New York, NY, USA: ACM, 2004, pp. 66–76.
- [19] Freedom House, "Freedom on the Net 2014."
- [20] L. Gao, "On inferring autonomous system relationships in the internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, Dec. 2001.
- [21] L. Gao and J. Rexford, "Stable internet routing without global coordination," *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 6, pp. 681–692, 2001.
- [22] P. Gill, M. Schapira, and S. Goldberg, "Modeling on quicksand: Dealing with the scarcity of ground truth in interdomain routing data," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 1, pp. 40–46, Jan. 2012.
- [23] V. Giotsas, M. Luckie, B. Huffaker, and k. claffy, "Inferring complex as relationships," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. New York, NY, USA: ACM, 2014, pp. 23–30.
- [24] A. Houmansadr and N. Borisov, "Swirl: A scalable watermark to detect correlated network flows," in *Proceedings of the Network and Distributed Security Symposium - NDSS'11*. Internet Society, February 2011.
- [25] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson, "Users get routed: Traffic correlation on tor by realistic adversaries," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 337–348.
- [26] J. Juen, A. Johnson, A. Das, N. Borisov, and M. Caesar, "Defending tor from network adversaries: A case study of network path prediction," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 1–17, 2015.
- [27] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. van Wesep, T. E. Anderson, and A. Krishnamurthy, "Reverse traceroute," in *Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2010, April 28-30, 2010, San Jose, CA, USA*, 2010, pp. 219–234.
- [28] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: an information plane for distributed services," in *OSDI*, 2006.
- [29] J. McLachlan and N. Hopper, "On the risks of serving whenever you surf: Vulnerabilities in Tor's blocking resistance design," in *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2009)*. ACM, November 2009.
- [30] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of tor," in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, ser. SP '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 183–195.
- [31] S. J. Murdoch and P. Zielinski, "Sampled traffic analysis by internet-exchange-level adversaries," in *Proceedings of the 7th International Conference on Privacy Enhancing Technologies*, ser. PET'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 167–183.
- [32] J. Qiu and L. Gao, "Cam04-4: As path inference by exploiting known as paths," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, Nov 2006, pp. 1–5.
- [33] B. Quoitin and S. Uhlig, "Modeling the routing of an autonomous system with c-bgp," *Netw. Mag. of Global Internetwkg.*, vol. 19, no. 6, pp. 12–19, Nov. 2005.
- [34] J. Rice, *Mathematical Statistics and Data Analysis*, ser. Duxbury advanced series. Duxbury Press, 1995, no. v. 1.
- [35] RIPE NCC, "RIPE atlas," <http://atlas.ripe.net>.
- [36] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Detecting prefix hijackings in the internet with argus," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 15–28.
- [37] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: Attacks and defenses," in *Proceedings of ESORICS 2006*, September 2006.
- [38] P. Smith, "Bgp routing table analysis," <http://thyme.apnic.net/>.
- [39] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal, "Raptor: Routing attacks on privacy in tor," pp. 271–286, Aug. 2015.
- [40] L. Vanbever, O. Li, J. Rexford, and P. Mittal, "Anonymity on quicksand: Using bgp to compromise tor," in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, ser. HotNets-XIII. New York, NY, USA: ACM, 2014, pp. 14:1–14:7.
- [41] C. Wacek, H. Tan, K. S. Bauer, and M. Sherr, "An empirical evaluation of relay selection in tor," in *20th Annual Network and Distributed System Security Symposium, NDSS 2013, San Diego, California, USA, February 24-27, 2013*, 2013.
- [42] T. M. P. website, "Tor project: Anonymity online," Available at <https://metrics.torproject.org>.
- [43] P. Winter and S. Lindsog, "How the Great Firewall of China is blocking Tor," in *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet (FOCI 2012)*, August 2012.