School of Informatics

Research Methods in Security, Privacy, and Trust
Routing Attacks and Defenses in Tor

s1997835

January 2020

Abstract

The anonymity network Tor is vulnerable to end-to-end correlation attacks, and fingerprinting attacks of encrypted traffic are improving. In order to launch these attacks, adversaries may attempt to actively influence routing through the Tor network or compromise relays or Internet infrastructure such that they lie on the path of Tor circuits. This literature review presents an overview of recent work on these attacks, as well as proposed defenses. Through this review, a number of desirable properties for alternative routing algorithms is compiled.

Date: Wednesday 8th January, 2020

Supervisor: Tariq Elahi

1 Introduction

Tor is perhaps the best-known anonymity network that is currently in operation. With an average of 2 million concurrent users at any given time [1], it is certainly the most popular. Clients route their traffic through three hops in the Tor network of over 7000 volunteer-run relays.

Research on routing in Tor can be broadly classified into either *(i)* routing attacks, in which the adversary attempts to exploit vulnerabilities in Tor or Internet infrastructure in order to increase their chance of observing both ends of a Tor circuit, or *(ii)* routing defenses, i.e. research on ways to route through the Tor network without being vulnerable to correlation attacks.

The initial seed papers for this review came from `freehaven.net`'s *Selected Papers in Anonymity* [2], a bibliography maintained by a co-founder of the Tor Project.

1.1 Background

1.1.1 Tor

Tor, or The Onion Router, gets its name from the fact that packets are wrapped in several layers of encryption. A Tor client, wishing to initiate a session, gets a list of all active Tor relays and builds a circuit to the destination site through three of these. The first relay in this circuit (the "guard") remains constant for up to nine months. The guard relay is the only relay to see the client's IP address. Upon receiving a packet, each relay will decrypt the outermost layer of encryption and forward it to the next relay. Thus, only the last relay in the circuit – the "exit relay" – knows the destination of the packet, and no single relay can see both the client and destination.

There are several classes of attack against Tor, but we are most interested in correlation attacks. Because Tor is a low-latency network (i.e. packets are forwarded to the next hop in the circuit immediately), an adversary who can observe traffic between the client and guard relay, as well as between the exit relay and destination, can correlate packets to work out which client is connecting to which destination. Correlation attacks achieve very high accuracies [3]. One of Tor's key mitigations against this is to make it costly for an adversary to control both the guard and exit relay with high probability.

The majority of the research covered in this literature review concerns itself with correlation attacks. However, it's worth noting that an attacker can gain a lot by observing the client-guard connection even without access to the exit-destination connection. Through fingerprinting attacks, an attacker may be able to gain insight into the sites visited by a client. Though fingerprinting attacks have their challenges, such as the extremely large number of websites, recent attacks on Tor are making progress and have few false positives [4].

1.1.2 Routing attacks and defenses

The relays in a Tor circuit are selected probabilistically, weighted by their available bandwidth (as well as other factors). Not only does this load-balance across the network, it also means that an adversary cannot simply set up a large number of cheap relays and thus have a high chance of having their relays chosen in circuits. To achieve this high chance of being selected, an adversary must supply a large proportion of Tor bandwidth – significantly more expensive than supplying a large number of relays.

Adversaries may look for other, more achievable ways to improve their chances of observing traffic on both ends of a Tor circuit. There is a small amount of research into application-level attacks (section 2.1), though the majority of recent work is concerned with BGP routing attacks (section 2.2). On the defensive side, there is a large body of research on routing to avoid certain regions or to avoid traversing the same network infrastructure on either end of the circuit (section 2.3). Finally, there are a number of more general analyses of guard placement attacks and guard selection algorithms (section 2.4).

2 Literature review

2.1 Application-layer attacks

One of the earlier routing attacks that exploited a vulnerability in Tor itself was found by Bauer et al. [5]. Their attack relied on the fact that there was no verification of a relay's advertised bandwidth, and thus they could deploy low-resource relays that falsely advertised a high bandwidth. These relays would then lie on an unfair proportion of Tor circuits.

A much more recent project is [6] by Kohls and Pöpper. They point out that recent DoS mitigations added to Tor relays [7] may present an attack vector. If an adversary can achieve a TCP man-in-the-middle attack between a client and guard relay, then they can flood the guard with traffic causing it to break the circuit. The client must then use another guard, increasing the probability of selecting a malicious relay run by the attacker. This research is ongoing.

In general, there is not much active research on application-layer attacks. This type of vulnerability is rare and (often) patched quickly.

2.2 BGP routing attacks

Tor circuits, like all Internet connections, are routed across Autonomous Systems (ASes). An AS is a set of IP prefixes that are controlled by a single entity. If a large number of Tor relays are hosted on a single AS, it means that the controlling entity has a higher chance of having visibility into both ends of a given Tor circuit.

Routing between ASes is done using the Border Gateway Protocol (BGP). BGP routing attacks occur when an AS announces a prefix that it does not own [8], leading other ASes to send it traffic that it should not see. The malicious AS can then either drop packets (if they announced a more specific prefix) or forward them to the correct destination while eavesdropping (if they announced an equally-specific prefix).

BGP routing attacks are a concern when creating Tor circuits. If an AS-level adversary can route Tor traffic through its own AS, then the probability of being able to perform correlation attacks increases. Sun *et al.* show how an adversary may go about this [9]. They point out that there are certain characteristics of BGP that make routing attacks easier: firstly, that AS paths may not be symmetric (i.e. the path of TCP packets may be different to the path for TCP acknowledgements), and secondly, that BGP routes change over time. Applying these characteristics to Tor gives rise to what the authors call RAPTOR attacks.

Their attack works in a scenario where the adversary aims to deanonymize a Tor client connecting to a given website. They assume that the adversary can already see traffic towards the website. The adversary must also first use existing attacks on Tor, e.g. [10], to learn the guard relay of the target. From there, the adversary can launch a BGP interception attack by

	Client ACK/ Server ACK	Client ACK/ Server Data	Client Data/ Server ACK	Client Data/ Server Data
Overall	96%	94%	96%	94%
False negative	4%	6%	4%	6%
False positive	0%	0%	0%	0%

Table 1: Correlation accuracy rate in Sun et al. [9]

advertising a more-specific IP prefix on their own malicious AS. Then they can correlate this traffic to deanonymize the client, and because they can forward packets to the true destination AS, they can do so without the client knowing. Sun et al. note that more than 90% of BGP prefixes that host Tor relays publish a prefix shorter than /24, meaning that they are vulnerable to these attacks (most ASes filter out announcements more specific than /24).

First, Sun et al. showed that correlation attacks have high accuracy regardless of whether an adversary can see direct TCP traffic or only TCP ACKs. They tested these attacks on clients downloading 100MB files over the live Tor network. Table 1 shows the accuracy rate of their correlations. They achieved a 95% accuracy on average.

In a second experiment, they attempted a more-specific BGP prefix attack, once again against clients downloading a 100MB file. In this setting, they achieve a 90% accuracy in deanonymizing clients. Though this is indeed a large accuracy, there are a number of requirements for an adversary to reach this point. They must have access to an AS, know the guard relay of the target client, and have access to traffic data from the website. It is also worth noting that downloading a 100MB file is perhaps rare in regular web browsing and represents a best-case scenario from the adversarial perspective. Finally, a more-specific prefix attack means that intercepted packets cannot be forwarded to the true destination (since the true destination would only send them back to the adversarial AS). All intercepted packets are lost. Though the attack was successful relatively quickly (in the order of minutes), a client is likely to notice that their packets are lost for several minutes.

2.2.1 Mitigations

A later paper by several of the same authors presents an in-depth list of suggestions for safeguards against these RAPTOR attacks [11]. They adapt an existing resiliency metric [12] to measure the Tor network's resilience to BGP hijacks. They then suggest some defenses: in particular, a guard selection algorithm that takes hijack resilience into account, as well as monitoring global BGP announcements to catch attempted hijacks.

They note that since the attacks in their original paper could achieve 90% accuracy in only five minutes, any detection of BGP hijacks must give actionable insights fast. The suggested guard selection algorithm chooses Tor guards probabilistically, weighted by both bandwidth and hijack resilience:

$$W(i) = \alpha \times R(i) + (1 - \alpha) \times B(i) \tag{1}$$

where $R(i)$ is the resilience, $B(i)$ is the available bandwidth, and α is a tunable parameter.

2.3 Avoidance routing

2.3.1 AS-aware routing

Partially motivated by RAPTOR attacks, there has been an increased research focus on routing algorithms that take ASes into account. While RAPTOR concerns attacks on BGP, an AS-level adversary may still be able to view traffic entering and leaving the Tor network in order to deanonymize clients, even without needing to launch active BGP attacks. The risk of AS-level adversaries was first highlighted by Feamster & Dingledine [13]. More recently, Juen et al. [14] compared traceroute measurements of Tor circuits' AS paths with the paths predicted by Qiu and Gao's algorithm, the state of the art in AS path prediction [15]. They found that these predicted AS paths achieved a average accuracy of only 60%.

At this point it is worth noting that traceroute measurements are not perfect. Reasons include that IPs returned by traceroute measurements may be from a different network interface than that of the receiver, and mapping an IP address to an AS is complicated by mistakes in WHOIS information. However, 90% of traceroute AS paths are consistent with advertised BGP paths [16]. In other words, Juen et al. conclusion was that existing ideas for AS-aware routing (generally using path prediction) may be better served by traceroute data.

This is still an area of controversy. Nithyanand et al. acknowledge the difficulties of path prediction but still rely on it for *Astoria*, an AS-aware Tor client [17]. This is justified by the use of a more recent prediction algorithm [18] that accurately predicts between 65% and 85% of measured paths [19].

To motivate Astoria, Nithyanand et al. show that 40% of Tor circuits are vulnerable to asymmetric correlation attacks (measured across 200 websites and 10 countries). When considering state-level adversaries who can snoop on traffic to or from all ASes in the country, they paint a grim picture: 82% of their tested websites served their main page over a vulnerable circuit. Their Astoria client generates a probability distribution over entry- and exit-relays that minimizes the chance of choosing vulnerable circuits. With this client, the percentage of sites vulnerable to a state-level adversary drops to 25%.

There is at least one attack against Astoria, and indeed any AS-aware client: depending on their location, some clients may not be able to build a safe circuit at all. In this situation, a relay-level adversary could simply deploy entry and exit relays that provide a "safe" circuit (i.e. paths to/from the client do not traverse the same AS). We can call this as *guard placement attack* as in [20]. Nithyanand et al.'s suggested solution to this is to only choose safe entry/exit relay pairs when there are more than some threshold n to choose from.

One downside of Astoria (as well as earlier AS-aware clients like LASTor [21]) is that they must know the destination of a request before constructing Tor circuits, and on-the-fly AS-aware routing is computationally expensive. Barton & Wright suggest DeNASA, a Tor client that can pre-construct AS-aware Tor circuits regardless of the destination [20]. It works by first limiting the ASes to a shortlist of eight *suspect ASes* that lie on a particularly high number of possible paths.

The selection of such a small subset of ASes, though perhaps counterintuitive, was based on empirical measurements. Barton & Wright simulated connections from the top 10 countries (in terms of Tor users) to the top 176 websites as ranked by Alexa. They discovered that with only a single suspect AS, 68% of clients build vulnerable streams, and with two, 91% of clients do so. However, when going from six to eight suspect ASes, the increase in clients building vulnerable streams was only 0.4%. Thus, including more suspect ASes in the pre-calculations would have a high performance cost but negligible impact.

This insight has not been used in more recent papers. However, it does point towards an occasionally-observed method in the literature: some papers only consider tier 1 ASes [11][20], i.e. ASes that do not have a provider and can reach any other AS without paying for it. In the forest of trees of ASes, tier 1 networks lie at the roots. Because each of the 17 current top-level ASes see very large proportions of Internet traffic, they are valuable targets for an AS-level adversary. However, this does not take into account the fact that all ASes are not equally vulnerable to compromise. While a nation state can resort to legal means to compromise an AS, a tier 1 network may have stronger incentives and legal teams to push back. They are also likely to have better security practices, making more covert attempts at compromise difficult. Whether or not there are tier-2 or tier-3 ASes that lie on the paths of either *(a)* a disproportionate number of likely Tor circuits, or *(b)* particularly interesting circuits has not been investigated.

Several studies consider ASes but not other pieces of Internet infrastructure like Internet Exchange Points (IXes) [13][17][20], even though many packets will also traverse these. One paper that does handle IXes is Johnson et al.'s proposal for a trust-aware path selection algorithm (”TAPS”) [22]. They present an attack against Astoria in which a Tor client visits an adversary-controlled website. This website can kick off a large number of requests to other adversary-controlled web servers (over new Tor circuits), letting the adversary see the pattern of exit relays chosen by the client. If the adversary controls web servers in 300 ASes, they found the average entropy of the client's AS to be less than 4 bits.

To avoid this problem, TAPS clusters all client locations and all destination locations. Locations in a given cluster are treated as if they were in the same location, thus putting many locations in the same anonymity set. Clustering is done by a ”trust-policy provider”, e.g. Tor's directory authorities. These trust-policy providers also compile a default trust policy – a probability distribution that one or more adversaries will compromise a given relay, AS, or IX. A default trust policy might assume that a single global adversary can compromise any target with some probability, or that every country is a separate adversary that can compromise all relays, ASes, and IXes within it with certainty. Finally, Johnson et al. note that a new routing algorithm should consider two cases: one where only some clients use it, and one long-term goal where all clients do. The first case should not put the early adopters at increased risk, and TAPS has variants for each case.

There is one other issue that most research does not explicitly handle. Johnson et al. discuss the fact that much AS-aware routing covers routing to a destination IP address [22]. However, the destination IP is rarely known by the client, so a DNS query is made (also from the exit relay). It is important that the DNS resolver used by the exit is also taken into account when choosing a circuit. This may incur a high performance overhead if the DNS response is not cached in the same AS as the exit relay and the circuit has to be rebuilt, or a new circuit has to be made for the DNS query.

Finally, Barton et al. present a new metric for anonymity called CLASI (Client AS Inference) [23]. This is a game between an adversary and a challenger. The adversary has a path simulator PS representing a model of the entire Tor network, as well as some path selection algorithm. The adversary gives PS to the challenger, who uses it to build a Tor circuit. The question is then whether the adversary can discover the client AS based only on the guard, middle, and exit relays, as well as the destination. With L as the client AS, L' as the adversary's guess, S_L as the set of all possible client ASes, and ϵ_S as information leakage of the client location:

$$Pr[L = L'] = \frac{1}{|S_L|} + \epsilon_S \tag{2}$$

In other words, if there is no leakage of the client AS ($\epsilon_S = 0$) then the adversary can only guess at random. Unlike metrics like time-to-first-compromise, this tells us something about a client's anonymity before they are deanonymized. In addition, Barton et al. show that CLASI captures anonymity loss in cases where entropy-based metrics like the Gini coefficient do not.

Barton et al. use CLASI to evaluate different parameter tunings of DeNASA, but the metric has not been applied to other proposed routing algorithms.

2.3.2 Geographical avoidance

Some Tor users may be particularly concerned about nation-state-level adversaries. For this reason, Tor clients can be set to avoid relays in some collection of countries. However, the fact that no relays are located in a given country does not mean that packets sent between them do not pass through the country. In 2017, Li, Herwig, and Levin presented DeTor, a system for provably avoiding geographic regions [24].

Their work is motivated by showing that under Tor's existing approach to geographic avoidance, 88% of circuits cannot provide guarantees that packets did not cross the given region. These guarantees come from speed-of-light measurements. Though there does exist research (e.g. [21][25]) into avoiding a list of client-provided ASes in Tor circuit (which can be used as a proxy for countries), AS relationships are dynamic and not necessarily publicized. This means that AS path inference only achieves 60% accuracy on average [15]. Thus, there may be value in a geographic avoidance that does not require knowledge of ASes.

With the insight that an adversary can lie about having a higher latency (by withholding packets) but they not about having a lower one, DeTor calculates lower bounds on the round-trip time (RTT) between relays. From here, we can calculate the shortest possible time to traverse the full Tor circuit as well as the excluded region. With $D_{min}(x_1, ..., x_n)$ as the minimum distance of the circuit x_1 to x_2, and so on, to x_n, we get

$$R_{min} = \frac{3}{2c} \cdot \min \begin{cases} 2 \cdot D_{min}(s, F, e, m, x, t) \\ 2 \cdot D_{min}(s, e, F, m, x, t) \\ 2 \cdot D_{min}(s, e, m, F, x, t) \\ 2 \cdot D_{min}(s, e, m, x, F, t) \end{cases} \tag{3}$$

where R_{min} is the shortest time required to traverse the circuit *and* the forbidden region, and s, e, m, x, t, F are the source, entry (or guard) relay, middle relay, exit relay, target, and forbidden region, respectively. The $\frac{3}{2c}$ comes from the fact that in practice, data on the Internet does not travel faster than two-thirds the speed of light (though no source is provided for this statement), and D_{min} is doubled to account for the round-trip time.

With the minimum R_{min} calculated, DeTor can check whether a packet provably avoided the forbidden region simply by comparing the measured end-to-end RTT R_{e2e}:

$$(1 + \delta) \cdot R_{2e2} < R_{min} \tag{4}$$

where δ is a tuning parameter. If this inequality holds, the packet could not possibly have entered the forbidden region. DeTor works by first getting the geographic coordinates of the source, destination, and all Tor relays. Then, the set of potential circuits is narrowed down by replace R_{e2e} in equation (4) with the lowest possible RTT.

In 2019, Kohls et al. highlight several oversights in DeTor [26]. For example, Tor relays are not uniformly distributed, and avoiding certain countries can cause unreasonable hits to performance. In addition, it assumes that routes are symmetric. This, as we saw in RAPTOR attacks [9], is not true.

The authors highlight a number of challenges that must be taken into account by any geographical avoidance system for Tor:

1. The Tor network (and the Internet) is diverse in the sense that connection lengths between relays are heavily dependent on network infrastructure. Time measurements should take this into account. In addition, some regions (particularly Europe and North America) have a disproportionally high number of relays.

2. Relays do not provide reliable information on their physical locations. External GeoIP databases can be used, but are untrusted.

3. An avoidance system must be realistic to deploy – i.e. it must have good sources of ground-truth information, it must not harm the existing security of Tor, and it cannot impose too heavy a performance burden.

Kohls et al. present a system, inspired by DeTor, that uses empirically measured RTTs between relays rather than theoretical limits based on GeoIP locations. They demonstrate experimentally this system, called TrilateraTor, is less restrictive in its circuit avoidance – it avoids 22% fewer circuits while still providing a high likelihood that a forbidden region was avoided, thus improving performance and security over DeTor in many cases.

2.4 Guard selection

In 2012, Tor clients would choose from a set of three guard nodes when forming circuits. The guards in these sets were rotated every 30-60 days. Elahi et al. showed, through a simulation-based approach, that guard rotation increased the probability that a client would choose a malicious guard (every rotation is an opportunity for an adversary to have their guard selected) [27]. If guards never rotated, only 10% of clients used a malicious guard over eight months. In contrast, using the rotation policy at the time, 14% of clients used the compromised guard in only three months. In 2014, Dingledine et al. proposed moving to clients having a single, long-term guard (with up to 9 months between rotations) [28]. This was later implemented in Tor clients.

Hayes & Danezis point out that if clients only use a single long-term guard, the anonymity sets of clients using a new guard are small [29]. As an alternative they flesh out the concept of guard sets, first suggested by Dingledine et al. [28]. Guard sets put all Tor guard relays into global buckets such that many users use the same guard set. These guards sets generally contain relays with similar bandwidths. When selecting a guard, it is chosen uniformly at random from a client's guard set. Even if an adversary can discover a client's entry guard, the client is in a large anonymity set. By representing guard sets and clients as a full binary tree, insertion and removal algorithms can be devised to handle relays coming online or going offline while preserving the stability of the relationship between guard sets and groups of users.

Hayes & Danezis present this as a first stage of the work on guard sets, and the design was later evolved by Imani, Barton, and Wright. They demonstrate that the initial model for guard sets is vulnerable to a relay-level adversary [30]. Such an adversary can wait until a target guard set is close to "breaking" (i.e. needing to add more guards), then artifically limit a malicious relay's bandwidth such that it has a high chance of being added to the guard set. In this dynamic setting, they found that an adversary controlling 1% of guard bandwidth could compromise 40% of guard sets in four months. Their proposed solution involved grouping guard sets by AS rather than by bandwidth; a guard selection system that can be combined with DeNASA [20] to achieve many of the same AS-aware routing properties.

AS-aware guard sets rely on the assumption that it is more difficult for an adversary to deploy relays in a given AS than it is to deploy relays with a given bandwidth. However, many Tor relays are centralized in the ASes of large cloud hosting providers. Though the proposed AS-aware guard sets can have several sets in a large AS, further analysis is needed to evaluate whether some ASes are at much higher risk than others. Hayes & Danezis' paper faces the same issue: they do not evaluate whether some guard sets (particularly those with lower-bandwidth relays) are at greater risk of compromise. The risk of compromise may be distributed non-uniformly across guard sets and thus clients.

Some of the most recent work on this topic comes from Wan et al. [31]. They formalise the notion of guard placement attacks and present a generic defense that provides provable security of any path selection algorithm. The key of this defense mechanism is that malicious guards should not be able to compromise a number of clients disproportionate to their bandwidth. To avoid this, the probability of selecting a guard should be bounded relative to the cost of bandwidth. The authors show that in the worst case, Counter-RAPTOR [11] lets a malicious guard get more than 5× the number of clients than it should according to its bandwidth, and for DeNASA [20] this ratio rises to 1490×. Their defense provides a parameter $\theta \geq 1$ which specifies a bound of guards' probability/cost ratios.

Finally, there is DPSelect, a differential-privacy-based guard selection algorithm [32]. Hanley et al. highlight the importance of considering the worst-case, as well as the average-case, privacy loss from proposed routing algorithms. Using the Max-Divergence metric they demonstrate that the worst-case privacy loss in Counter-RAPTOR is significantly greater than in the average case. They modify Counter-RAPTOR to use a bounded worst-case Max-Divergence. However, they note that Counter-RAPTOR's resilience metric (see equation 1) is an average resilience over all possible adversaries. A more detailed analysis might come from a metric that considers a particular adversary, or the worst-case adversary for a particular client.

3 Conclusion

Though there is a forthcoming paper on application-layer routing attacks [6], there is not much research happening in this area. AS-level attacks and defenses are much more widely studied. RAPTOR attacks that apply BGP routing attacks to Tor [9] highlighted the need to consider asymmetric AS paths. There is a large body of work on avoidance routing, though as of yet no solution that has all the desirable properties that have been pointed out. Based on the above, these properties are as follows:

1. Realistic deployability, considering the fact that clients do not upgrade all at once [22][26]
2. Allows load balancing through the Tor network [17][26]

3. Does not heavily impact performance, or at least allows a configurable security vs. performance parameter [11][17][22][24]

4. Allows the user to specify countries to avoid, not just ASes [22][24]

5. Awareness of asymmetric AS paths [9]

6. Analyzes the worst-case, as well as the average-case privacy loss [32]

7. Takes ASes as well as IXes and relay families into account [14][22]

8. Takes DNS queries into account [22]

9. Considers the fact that a standard Tor client will create many circuits, not just one [22]

10. No reliance on an external, trusted source of GeoIP data [26]

Future work should consider these properties and evaluate the relative importance of each.

References

[1] The Tor Project. Users - Tor metrics. `https://metrics.torproject.org/userstats-relay-country.html`. Nov. 2019.

[2] Free Haven Project. Selected Papers in Anonymity. `https://www.freehaven.net/anonbib`. Oct. 2019.

[3] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. DeepCorr: strong flow correlation attacks on Tor using deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1962–1976. ACM, 2018.

[4] Tobias Pulls and Rasmus Dahlberg. Website fingerprinting with website oracles. *Proceedings on Privacy Enhancing Technologies*, 1:235–255, 2020.

[5] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Low-resource routing attacks against Tor. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 11–20. ACM, 2007.

[6] Katharina Kohls and Christina Pöpper. Poster: Application-layer routing attacks on Tor. In *40th Annual IEEE Symposium on Security and Privacy*, 2019.

[7] dgoulet. Denial of service mitigation subsystem. `https://trac.torproject.org/projects/tor/ticket/24902#02`. Jan. 2020.

[8] Ola Nordström and Constantinos Dovrolis. Beware of BGP attacks. *ACM SIGCOMM Computer Communication Review*, 34(2):1–8, 2004.

[9] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. RAPTOR: Routing attacks on privacy in Tor. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 271–286, 2015.

[10] Steven J Murdoch and George Danezis. Low-cost traffic analysis of Tor. In *2005 IEEE Symposium on Security and Privacy (S&P'05)*, pages 183–195. IEEE, 2005.

[11] Yixin Sun, Anne Edmundson, Nick Feamster, Mung Chiang, and Prateek Mittal. Counter-RAPTOR: Safeguarding Tor against active routing attacks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 977–992. IEEE, 2017.

[12] Mohit Lad, Ricardo Oliveira, Beichuan Zhang, and Lixia Zhang. Understanding resiliency of internet topology against prefix hijack attacks. In *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*, pages 368–377. IEEE, 2007.

[13] Nick Feamster and Roger Dingledine. Location diversity in anonymity networks. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, pages 66–76. ACM, 2004.

[14] Joshua Juen, Aaron Johnson, Anupam Das, Nikita Borisov, and Matthew Caesar. Defending Tor from network adversaries: A case study of network path prediction. *Proceedings on Privacy Enhancing Technologies*, 2015(2):171–187, 2015.

[15] Jian Qiu and Lixin Gao. Cam04-4: AS path inference by exploiting known AS paths. In *IEEE Globecom 2006*, pages 1–5. IEEE, 2006.

[16] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy H Katz. Towards an accurate AS-level traceroute tool. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 365–378. ACM, 2003.

[17] Rishab Nithyanand, Oleksii Starov, Adva Zair, Phillipa Gill, and Michael Schapira. Measuring and mitigating AS-level adversaries against Tor. *arXiv preprint arXiv:1505.05173*, 2015.

[18] Phillipa Gill, Michael Schapira, and Sharon Goldberg. Modeling on quicksand: Dealing with the scarcity of ground truth in interdomain routing data. *ACM SIGCOMM Computer Communication Review*, 42(1):40–46, 2012.

[19] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Ítalo Cunha, Phillipa Gill, and Ethan Katz-Bassett. Investigating interdomain routing policies in the wild. In *Proceedings of the 2015 Internet Measurement Conference*, pages 71–77. ACM, 2015.

[20] Armon Barton and Matthew Wright. DeNASA: Destination-naive AS-awareness in anonymous communications. *Proceedings on Privacy Enhancing Technologies*, 2016(4):356–372, 2016.

[21] Masoud Akhoondi, Curtis Yu, and Harsha V Madhyastha. LASTor: A low-latency AS-aware Tor client. In *2012 IEEE Symposium on Security and Privacy*, pages 476–490. IEEE, 2012.

[22] Aaron Johnson, Rob Jansen, Aaron D Jaggard, Joan Feigenbaum, and Paul Syverson. Avoiding the man on the wire: Improving Tor's security with trust-aware path selection. *arXiv preprint arXiv:1511.05453*, 2015.

[23] Armon Barton, Matthew Wright, Jiang Ming, and Mohsen Imani. Towards predicting efficient and anonymous Tor circuits. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 429–444, 2018.

[24] Zhihao Li, Stephen Herwig, and Dave Levin. DeTor: Provably avoiding geographic regions in Tor. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 343–359, 2017.

[25] Matthew Edman and Paul Syverson. AS-awareness in Tor path selection. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 380–389. ACM, 2009.

[26] Katharina Kohls, Kai Jansen, David Rupprecht, Thorsten Holz, and Christina Pöpper. On the challenges of geographical avoidance for Tor. In *NDSS*, 2019.

[27] Tariq Elahi, Kevin Bauer, Mashael AlSabah, Roger Dingledine, and Ian Goldberg. Changing of the guards: A framework for understanding and improving entry guard selection in Tor. In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, pages 43–54. ACM, 2012.

[28] Roger Dingledine, Nicholas Hopper, George Kadianakis, and Nick Mathewson. One fast guard for life (or 9 months). In *7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2014)*, 2014.

[29] Jamie Hayes and George Danezis. Guard sets for onion routing. volume 2015, 06 2015.

[30] Mohsen Imani, Armon Barton, and Matthew Wright. Guard sets in Tor using AS relationships. *Proceedings on Privacy Enhancing Technologies*, 2018(1):145–165, 2018.

[31] Gerry Wan, Aaron Johnson, Ryan Wails, Sameer Wagh, and Prateek Mittal. Guard placement attacks on path selection algorithms for Tor. *Proceedings on Privacy Enhancing Technologies*, 2019(4):272–291, 2019.

[32] Hans Hanley, Yixin Sun, Sameer Wagh, and Prateek Mittal. DPSelect: a differential privacy based guard relay selection algorithm for Tor. *Proceedings on Privacy Enhancing Technologies*, 2019(2):166–186, 2019.