We Built This Circuit:
Exploring Threat Vectors in Circuit Establishment in Tor

Theodor Schnitzler*, Christina Pöpper†, Markus Dürmuth*, and Katharina Kohls‡
*Ruhr-Universität Bochum, Germany †New York University Abu Dhabi, UAE ‡ Radboud University, Netherlands
theodor.schnitzler@rub.de christina.poepper@nyu.edu markus.duermuth@rub.de kkohls@cs.ru.nl

***Abstract*—Traffic analysis attacks against the Tor network are a persisting threat to the anonymity of its users. The technical capabilities of attacks against encrypted Internet traffic have come a long way. Although the current state-of-the-art predicts high precision and accuracy for website fingerprinting and end-to-end confirmation, the concepts of these attacks often solely focus on their *technical* capabilities and ignore the *operational* requirements that are mandatory to get access to transmissions. In this work, we introduce three novel stepping-stone attacks that enable an adversary to (i) gain additional information about monitored connections, (ii) manipulate the Tor connection build-up, and (iii) conduct a targeted Denial-of-Service attack within the Tor infrastructure. All attacks exploit core *defensive* features of Tor and, consequently, are hard to patch. At the same time, our attacks are in line with standard attacker models for traffic analysis attacks. We demonstrate the feasibility of all three attacks in simulations and empirical case studies and emphasize their pivotal role in preparing a *realistic* setting for end-to-end confirmation attacks.**

1. Introduction

With more than two million daily users, Tor [44] remains the most prominent anonymity system worldwide. Tor can serve everyday use cases with low-latency requirements and provides a fair amount of protection for user identities. However, this trade-off between performance and security comes at the expense of being vulnerable to traffic analysis attacks [13], [27]. In those attacks, the adversary uses metadata leaks from different points of a connection, eventually de-anonymizing Tor users.

In the last two decades, attacks against encrypted Internet traffic have gotten more sophisticated, beginning with the first attack on SSL traffic [9], [41] and currently evolving around automated deep-learning attacks [31], [37]. In this context, we focus our research on end-to-end (E2E) confirmation attacks. In an E2E confirmation, the adversary monitors traffic between the client and entry relay (related to the user IP address) and between the exit and the server (related to the content). Similarities between the entry and exit transmissions allow finding related transmissions, which enables the adversary to match an IP address with the accessed server. Numerous passive [10], [30]–[32] and active [16], [17], [36] traffic analysis attacks indicate the perspective of a persisting attack vector that affects past, present, and future systems.

Among these attacks, we find convincing technical concepts approaching almost 100 % success rates for the de-anonymization of related streams [31]. At the same time, all of these attacks ignore the operational requirements for getting access to transmissions. That is, the attack can only succeed in case the adversary is able to monitor both endpoints involved in the connection. As Tor has a worldwide infrastructure of 6,000 to 7,000 voluntarily operated relays, this results in high resource requirements for monitoring candidate connections or nodes [33], [38].

In this context, *long-term* evaluations of end-to-end confirmation in practice have shown that adversaries controlling specific Autonomous Systems (ASes) or Internet exchange points (IXPs) can de-anonymize individual circuits of 100 % of users within a three-month period [23] and that compromise can be more effective with Border Gateway Protocol (BGP) level routing attacks [42]. However, the feasibility of end-to-end confirmation attacks on a per-case basis remains a blind spot, and we must assume enormous resource requirements for a naïve monitoring and analysis of AS- or nation-state adversaries.

In this work, we introduce three *stepping-stone* attacks that tackle the operational limitations of state-of-the-art E2E confirmation attacks and provide the adversary information about monitored connections as well as tools to interfere with the connection build-up procedure in Tor.

To remain in line with common attacker models in the context of traffic analysis attacks, we design our stepping stones in a way that does not introduce additional requirements or constraints for the adversary. To this end, we integrate our attacks into *defensive* features of Tor's circuit establishment procedure, making them a hard-to-counter "standard feature" of current Tor versions. This includes (i) inherent characteristics of the circuit establishment such as relay selection as well as (ii) mechanisms that have been introduced for protection purposes. For the latter, we focus on the `nTor` handshake ensuring *onion encryption* and *denial-of-service mitigation* that protects relays from being stressed. Figure 1 provides an overview of the systematic security analysis of these characteristics, which leads us to three stepping-stone attacks: *Exit Prediction*, *Circuit Replacement*, and *Multi-Target DoS*.

Exit Prediction provides additional information about the monitored connections, which helps to minimize the attack effort for non-global adversaries. For example, a nation-state adversary can conduct the Exit Prediction attack to check whether the exit traffic of a circuit passes through a country under control and, eventually, would lead to a successful E2E confirmation. This information about connections introduces a significant advantage over uninformed attacks in which all monitored traffic must be analyzed while related traces might not even be part

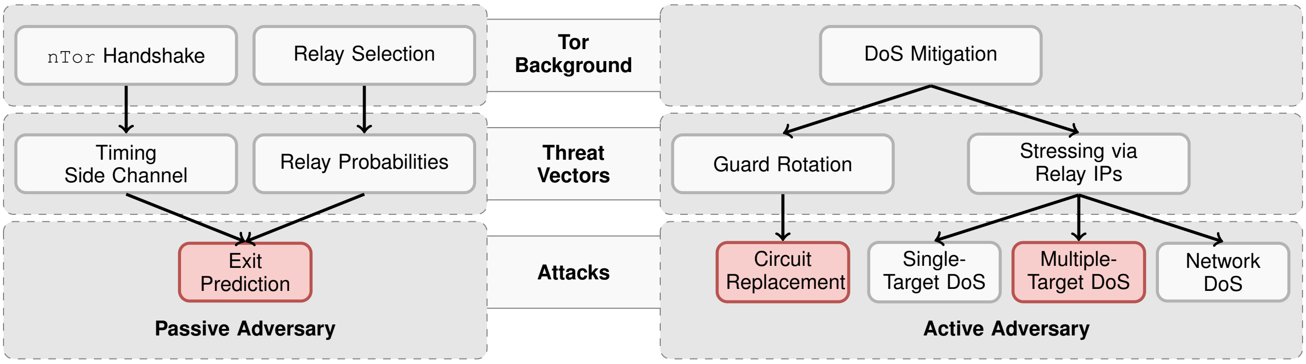

Figure 1. Structural overview of threat vectors and related attacks rooted in essential features and protection mechanisms related to Tor's circuit establishment procedure (top layer). For three of the attacks (highlighted in red) we provide empirical case studies.

of the monitored connections. In an empirical simulation study, we analyze the prediction capabilities of different probabilistic models and further analyze how an adversary performs with and without the Exit Prediction stepping-stone. Our experiments show that, depending on the individual infrastructure of a country, the exit prediction ranks the correct relay of a circuit within the top 1 % to 18 % of all possible relays. This drastically reduces the required effort for an attack, as only a fraction of traffic needs to be analyzed.

Circuit Replacement and Multi-Target DoS both exploit the Denial-of-Service mitigation within Tor. Circuit Replacement allows an adversary to interfere with the guard set of a user by stressing the primary guard. This triggers the DoS mitigation and forces the user's client to switch to the next guard in the set, eventually introducing a new relay location and transmission path. This local application-layer routing attack allows an adversary to manipulate a circuit in case the original connection does not allow monitoring traffic. This introduces additional attempts to access the connection endpoints of a user. Our experiments show that the circuit replacement provides an improvement of up to 33 % for adversaries that could not access traffic before the replacement attack.

Multi-Target DoS exploits the same DoS mitigation from inside the Tor network. Due to an implementation characteristic of the DoS mitigation, excessive connection attempts from *inside* the network are not blocked. This allows an adversary to stress single or multiple nodes in the infrastructure, which creates local failure or even complete intersections of network areas. Again, this can be used as a stepping stone for traffic analysis attacks, since it provides another tool to manipulate the connections within Tor. Our results show that individual relays can be disabled for one hour for around $ 20.

In short, the main contributions of our work are:

- We identify threat vectors rooted in core mechanisms and defensive features that are part of Tor's circuit establishment procedure.
- We analyze the characteristics and technical requirements for three attacks exploiting these threat vectors and facilitating traffic-analysis attacks.
- We use measurements of the live Tor network for simulation studies demonstrating the impact of the three mentioned attacks and their consequences for subsequent traffic analyses. Our experiments provide insights into case studies in real-world scenarios without harming real Tor users.

2. Tor Background

Connections through the Tor network use *circuits* that consist of three relays, i. e., an entry guard connecting to the user's Tor client, an exit relay connecting to the destination of the connection, and a middle relay as the link between the entry and the exit. The circuits are built during the bootstrap procedure in the client start-up of Tor and are ready-to-use for new connections.

In this section, we describe characteristics connected to the circuit establishment procedures in Tor, as well as defensive mechanisms that Tor has put in place to ensure user anonymity and to safeguard the stability of its network infrastructure.

2.1. `nTor` Handshakes

The circuit establishment procedure involves multiple layers of encryption. The Tor client conducts three key establishments handshakes with the entry (E), middle (M), and exit (X) relays, as illustrated in Figure 2. To protect transmitted communication contents from relays in the circuit, the client follows the `nTor` protocol to establish individual layers of *onion encryption* with each relay separately. Since the client's identity must not be revealed to the middle and exit, connections with relays positioned later in the circuit transit through hop-wise encrypted TCP connections with previous relays. The `nTor` protocol ensures that exchanged messages remain secure while preserving the client's anonymity.

2.2. Relay Selection

To keep track of all voluntarily contributed nodes, Tor uses a distributed consensus that consists of periodical votes on the existing infrastructure. This publicly available consensus assures easy access to the status of all available nodes. For a new circuit to be established, the client picks relays from Tor's worldwide infrastructure and focuses its choice mainly on the advertised bandwidth a node can offer in order to reach a fair distribution of traffic. The

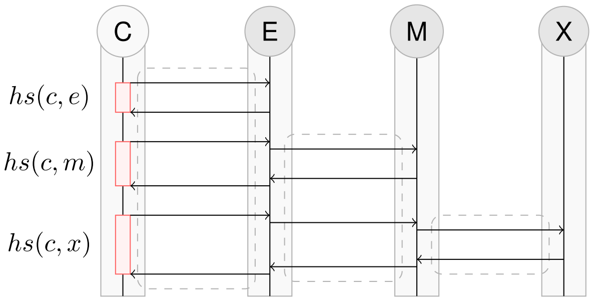

Figure 2. `nTor` Handshake. The client (C) establishes keys with each relay in the circuit, in the depicted order. Key establishments with posterior relays transit through the circuit, multiplexed over separated TLS connections (dashed areas) between pairs of relays to keep the client identity private towards the middle (M) and exit (X) relays.

decentralized Tor infrastructure is backed by individuals and organizations worldwide, voluntarily operating relays with diverse amounts of resources they can commit. Thus, the geographical distributions of relays and the available bandwidth are subject to the capabilities of voluntary contributors. As illustrated in Figure 3, relay numbers per country are diverse with highest relay densities in central Europe (e. g., Germany, Netherlands, France) and the US. Especially in the Netherlands, there are 240 relays in an area of 500 square kilometers around the city of Amsterdam, providing 33 Gbit/s relay bandwidth in total, which comprises about 5 % of the overall Tor bandwidth.

There are further constraints that contribute to the composition of a Tor circuit by default, even though almost all of them can be manually overwritten by the client. By default, two relays from the same family (as specified in their descriptions) or residing in the same /16 subnet (i. e., their IPv4 addresses must not be equal in the first two blocks) are not selected for a single circuit.

Guards and Exits. The entry and exit relays in a circuit are particularly crucial for a circuit's security, as they directly communicate with the client (entry) and destination server (exit). The distinctive roles of both nodes are taken into account by assigning flags for relays that might serve in one of these critical positions of a circuit. While both flags are assigned after satisfying a series of requirements, entry guards are additionally organized in client-specific guard sets [14].

When a relay receives the `general` guard flag, the Tor client can sample it to become part of the client-specific guard set. Within these guard sets, the client keeps track of the connection status. This results in a sampled guard set of 15 relays on average, of which one to three relays have the status `up` and will be used in a circuit. Each of the `up` relays is assigned an index resembling the internal priority, i. e., the highest priority entry guard will be used in all general-purpose circuits if possible. The client switches to other primary guards of the set when the highest priority node is unavailable. The client creates a guard set once in the bootstrap procedure (if none is given) and updates nodes after a lifetime of several months [11].

The option to use guard sets can be changed by each client, allowing them to also use non-guard-flagged relays as entry nodes. In contrast, for circuits with traffic leaving the Tor network, only exit-flagged relays can be used in the exit position. The decision about allowing exit traffic is made by the relay provider. That is, relays are not picked at random but following a deterministic procedure.

Consequently, the actual composition of a circuit and its transmission characteristics depend on relay performance and geographical features. All information about available relays, their flags, or their advertised bandwidth is accessible from the consensus files that Tor updates in an hourly schedule using a decentralized voting infrastructure.

2.3. DoS Mitigation

One major threat to the Tor infrastructure are Denial-of-Service (DoS) attacks, in which the adversary floods relays through bursts of circuit and connection attempts. Since version `0.2.4.18-rc` [20] released in 2013, Tor implements DoS mitigation features that protect an entry relay from such excessive requests coming from a single IP address.

We focus on DoS mitigation parameters targetting both the number of circuits that can be created from a single IP address and the number of parallel connections from a single IP address and defining consequences if the specified limits are exceeded (cf. Listing 1).

Listing 1. Denial-of-Service Mitigation Options

```
DoSCircuitCreationEnabled 0|1|auto
DoSCircuitCreationMinConnections NUM
DoSCircuitCreationRate NUM
DoSCircuitCreationBurst NUM
DoSCircuitCreationDefenseType NUM
DoSCircuitCreationDefenseTimePeriod N

DoSConnectionEnabled 0|1|auto
DoSConnectionMaxConcurrentCount NUM
DoSConnectionDefenseType NUM
```

The `Enabled` parameters define whether creating new circuits or establishing new connections is currently enabled. The `Circuit` options cover circuit creation requests, i. e., the creation rate, and the creation burst that define the allowed number of circuit creations per second and the maximum burst, respectively. The `MinConnections` defines the number of concurrent connections that must be present to trigger the mitigation feature and, eventually, the blocking of an IP address. For the `Connection` features, the maximum number of connections specifies the number of concurrent connections that are allowed from a single IP address at a time and closes new connections if exceeded. In combination, the `Circuit` and `Connection` features block excessive requests and mark an IP address for the time defined in the `Defense` parameters. In case the relay provider does not specify any value for these features, the default setup still assures an active DoS mitigation.

3. Threat Vectors

In this section, we sketch how we exploit the presented characteristics and defensive mechanisms in Tor. The critical problem with the threats that we will present

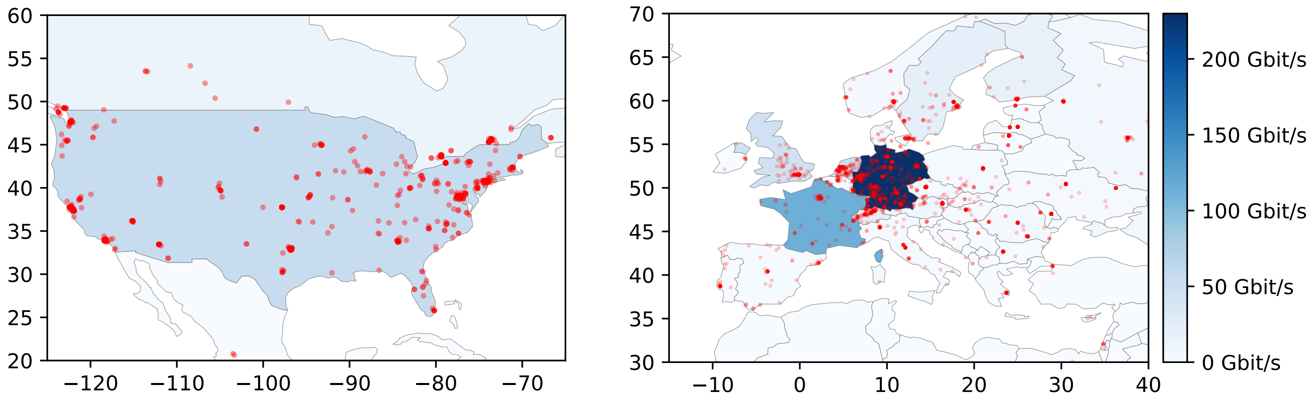

Figure 3. Distribution of Tor relays in North America and Europe. Colors in areas denote total relay bandwidth per country (darker color implies higher bandwidth), relay locations are marked with red dots.

is that they are inherent to the fundamental concepts behind Tor, i. e., they cannot be mitigated without in-depth changes of the way how Tor works. For each threat vector, we describe how it is rooted within Tor and conduct preliminary experiments. This provides the baseline for the introduction of the three stepping-stone attacks.

3.1. Timing Side Channel

Similar to contexts like GPS, we can assume that the propagation time of signals between two nodes in a network relates to the traveled physical distance between these nodes. Given a suitable timing side channel, an adversary can make use of the timing relations and determine the geographical areas that relays are likely located in.

The cryptographic key establishment in Tor's circuit build-up procedure provides such a timing side channel. For such a circuit build-up, the Tor client and each relay in the circuit exchange messages as part of the `nTor` handshake protocol (cf. Section 2.1). Each message comprises a timing side channel. For example, observing the handshake between the client and the entry relay reveals the end-to-end round trip time between these two nodes.

In the following steps, we benefit from the fact that each new handshake message must follow the circuit infrastructure. More precisely, the handshake between client and middle includes the connection between client and entry, of which we already know the individual *RTT(c,e)*. This enables us to approximate the transmission time between entry and middle relay, as *RTT(c,m) - RTT(c,e)* = *RTT(e,m)*. Following this principle, we can derive the transmission times of all three *individual* hops *RTT(c,e)*, *RTT(e,m)*, *RTT(m,x)* from the combination of timings.

We conduct a series of preliminary experiments to analyze the practicality of the handshake timing side channel to be later used as a stepping-stone for end-to-end confirmation attacks. In particular, we analyze how transmission characteristics depend on traveled distances, and we measure to what extent the cryptographic operations in the handshake protocol introduce overhead into the observable end-to-end timings.

Transmission Characteristics. We analyze the propagation times of the empirical handshake data derived from 84,500 weighted circuit establishments (by weighted we refer to the standard Tor circuit build-up, i. e., we do not interfere with the selection of relays). Figure 4 visualizes the empirical handshake timing data ($n = 5000$, scatter) by distance between two hops approximated by a polynomial fit (lines). While we generally see higher handshake timings for longer distances, we also see timings scattered a lot around similar distances, with exit timings being higher (i. e., resembling lower transmission speeds) than entry or middle timings. The gap of $3{,}000\,\mathrm{km}$ to $5{,}000\,\mathrm{km}$ is caused by the static trans-Atlantic connection between the North America and Europe, e. g., inter-European or inter-US connections are either shorter (no transit) or longer (transit and distance to relays).

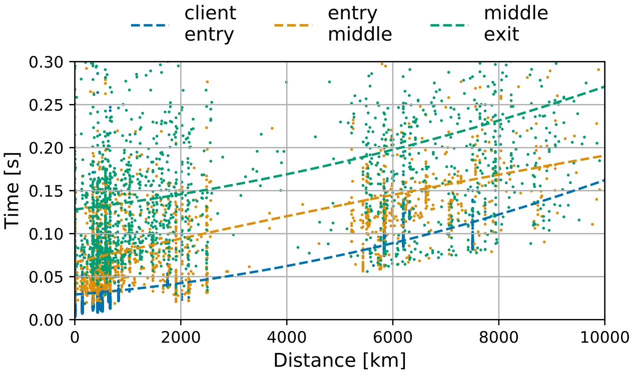

Figure 4. Distribution of handshake times by distance between pairs of hops. Data derived from a random sample ($n = 5000$) of circuit establishments measurements.

Static Overhead. The observed end-to-end round trip times comprise transmission timings between the relays and also include a computational overhead for the key establishment procedures. We analyze this overhead by running a patched Tor relay that records the time delta for *processing* the handshake, i. e., the time difference between the start and the end of the server handshake. Over a period of 12 hours we observed 138,000 key establishment timings on the server side with a median of $22\,\mu\mathrm{s}$ ($\sigma = 44\,\mu\mathrm{s}$). As we see in the analysis of handshake timings, this overhead is negligible.

3.2. Relay Probabilities

When a Tor client establishes a new circuit, relay choices are not uniformly random but depend on several factors and are mainly driven by the advertised bandwidth of a node. Therefore, nodes have different probabilities to become a relay in the new circuit.

We describe two statistical approaches that can be used to assign probabilities on different levels, e. g., for individual nodes to become part of a new circuit, or for a circuit to contain relays located in particular countries.

Relay Selection-based Estimates. The selection of relays in Tor is mainly driven by a node's advertised bandwidth, information that is publicly accessible in Tor's consensus. Therefore, we can approximately determine the probability P_{bw} for a node x to be selected by considering its bandwidth as a fraction of the overall available bandwidth contributed by all nodes in the consensus:

$$P_{bw}(x) = bw(x) * \frac{1}{\sum_{i=1}^{n} bw(i)} \tag{1}$$

These calculations can be performed in real time and do not require any preparation other than downloading the hourly updated current Tor consensus.

Equally, we cannot only determine probabilities for individual relays, but also for groups of relays that are part of a specific autonomous system (AS) operated by a particular network provider, or relays located in a specific geographic area, each depending on the information available in the consensus.

When determining these estimates, we also consider constraints in relay selection, more precisely, we exclude all relays that are in the same relay family or in the same /16 subnet as one of the other relays in the circuit [45].

Timing-based Estimates. We assume that, even if there is no direct relation between timings and traveled distances, transmission times between two specific relays remain similar over time, i. e., that we can observe the handshakes of a newly established circuit and identify the relays involved by comparing the handshake timings with previously collected data for the same or a similar connection.

Determining relay probabilities based on timings requires collecting a sufficiently large sample of timing data first, to determine the full range of likely timings for different connections. We can then generate empirical timing distributions for these connections, e. g., between pairs of relays or relay areas. Upon measuring the handshake time of a newly established circuit, we can then extract a probability P_t for each relay x by considering the timing distribution for the respective connection.

While the overall idea here is conceptually the same as for relay selection-based estimates – assigning each node an individual probability for being involved in a newly established circuit – timing-based estimates require a lot more preparation.

3.3. Guard Rotation

A Tor client establishing circuits constantly uses the same guard relay in the entry position of all circuits, and usually does not change its behavior as long as the guard is available, i. e., changes it only in exceptional cases. This behavior is considered a security measure to protect the client and to reduce the risk of being exposed to malicious relays, since the entry connection is a critical point for client privacy.

The DoS mitigation features implemented in Tor relays (cf. Section 2.3) use client IP addresses as identifiers and do not allow any more connections or circuits from a specific IP, as soon as the limits specified for the mitigation features are exceeded. Since this mechanism is purely IP-based, it can also be triggered by excessive requests from entities pretending to possess a specific IP address. In this case, a client can be forced to switch from its primary guard to another relay from its guard set, without the change being necessary, and without the client being aware of the situation. Consequently, the client must create a new set of circuits.

We conduct a preliminary experiment to validate the presumed behavior of a client switching its main guard upon exceeding the DoS mitigation limits.

Triggering DoS Mitigation. We verify the client's behavior by stressing our own Tor relay. We run our client on a local machine and set up our own relay `Torben` running on a remote virtual machine instance.

We first assure that `Torben` is the primary guard in the guard set of our client, such that it is picked as entry relay in the circuits we create. Upon starting the client, we drop all guards of the current guard set and manually add `Torben` to the empty set, rendering it the only guard. Shutting down the Tor client and restarting it adds new guards to the set. Besides our own relay, 2 to 3 additional entry guards with the status `up` are sampled in the list.

We run in total 20 Tor instances on our client, one of which uses the manipulated guard set with `Torben` as the primary guard; all others are used for stressing the DoS features. We first check our client's functionality, i. e., that it can build and use general-purpose and internal circuits. Both are satisfied when our relay shows up in the guard set, and Tor prepared a series of ready-to-use three-hop circuits with `Torben` in the entry position. In the next step, we build 20 circuits in each of the Tor instances with `Torben` in the entry position.

In the first Tor instance, all circuits of the initial build-up remain present and decay over time when their lifetime passes. It is impossible to build *new* circuits with `Torben` in the entry guard position. As older circuits disappear over time, Tor starts to build new circuits that now use one of the other relays of the guard set. These circuits show up in the circuit list and can also be used to attach streams for transferring data. Therefore, we have successfully triggered our first client instance to switch to another entry guard as a consequence of triggering the primary entry guard's DoS mitigation.

3.4. Stressing via Relay IPs

Even though Tor has established techniques to mitigate denial-of-service attacks, its mitigation features have one specific characteristic: They can only be triggered from *unknown* IP addresses, i. e., nodes that are not part of the consensus. For IP addresses that are part of the

Tor network, DoS mitigation features, as described in Section 2.3, do not apply. Therefore, targeted denial-of-service attacks affecting the stability and availability of Tor are successfully prevented when conducted from the outside but still remain possible from within Tor. However, exploiting this threat vector is limited to particular actors - for those who can either spoof valid IP addresses used by Tor nodes, or those who actually possess and control these address spaces.

Analyzing one exemplified consensus, we find 976 different Autonomous System (AS) operators that serve approximately 6,700 relays of the Tor infrastructure. While many of the operators only serve 1 to 10 relays, larger AS regions include up to 746 (OVH SAS, 110 Gbit/s total bandwidth) or 409 (Hetzner Online GmbH, 100 Gbit/s total bandwidth) relays within their area of control. Consequently, without depending on additional hardware, a malicious provider can conduct a Distributed DoS attack using all IP addresses of relays falling into their AS area. In other words, the adversary can stress Tor relays without triggering their DoS mitigation, simply because IP addresses listed in the consensus are excluded from the mitigation.

4. Attack Concepts

Given the threat vectors of Section 3, we now introduce the specific attack concepts and how they support an adversary in conducting a end-to-end confirmation attack. To this end, we first introduce different models for the *operational* capabilities of an adversary. We then introduce the detailed concepts of the three stepping-stone attacks, which we later analyze in case studies of practical attack scenarios (cf. Section 5).

4.1. Attacker Models

In the context of network attacks, an adversary with access to transmissions on the Internet (IP) or Transport Layer (TCP, UDP) can conduct a series of active and passive attacks. The chance of being successful mainly depends on the operational capabilities of the adversary. For example, a local adversary has access to the same type of information as a global adversary; however, the *amount* of information differs significantly. We specify three operational classes of adversaries that define the possible scope of an attack. For each attack concept, we extend this by specific technical capabilities.

Global Adversary. The global adversary can access all nodes in the network infrastructure and conduct arbitrary measurements.

Autonomous Systems and Nation States. An autonomous system can access all traffic routed through its service area. Depending on the centrality of a country's infrastructure, this can vary from multiple provider areas to one dominant provider operating the majority of connections. The nation-state adversary is an operational concept in which we assume a powerful entity that can request access to traffic in arbitrary points of a country.

Local Adversary. This adversary has access to traffic in a local network, e. g., uses the same access point in a public WiFi, and can monitor all traffic of this network.

4.2. Exit Prediction

The exit prediction provides the adversary with additional information about a connection. More precisely, we assign all relay candidates within the Tor infrastructure a probability for being in the exit position of a circuit. We do this by combining the timing side channel (cf. Section 3.1) with probabilities derived from consensus statistics. The outcome of an exit prediction is the list of all exit relays ranked in order of likelihood for being in the exit position of a single circuit. Upon receiving a relay ranking for a specific exit prediction, an adversary can determine whether subsequent traffic analyses are promising, depending on how the relays under their control (i. e., those they are able to observe) are ranked in the prediction. That is, the exit prediction can serve as an indicator of whether an attempted end-to-end confirmation can be successful, eventually helping the adversary to save resources. It serves as an optional pre-analysis step that does not require additional technical or operational capabilities.

We first describe the concept behind and underlying assumptions of the attack and provide an empirical evaluation of the exit prediction using a simulation study in Section 5.1.

4.2.1. Technical Attacker Capabilities. The technical and operational requirements for the exit prediction are included in all possible attacker models of an end-to-end confirmation: To exploit the timing side channel of a specific client, the adversary either requires access to the client's entry connection or to Tor relays that are flagged as Guard. This can be achieved by a locally restricted adversary, the minimum adversary for an end-to-end confirmation (cf. Section 4.1).

4.2.2. Concept. Taking up on the *Relay Probabilities* as threat vector (cf. Section 3.2), the procedure for generating a bandwidth-based prediction ranking here is straightforward. The outcome of a prediction is simply a ranking of exit relays, with probabilities determined by their individual bandwidth fraction (cf. Equation 1). However, this is not a suitable mechanism to provide an adversary information on their chances for subsequent attacks. The prediction remains the same unless the consensus is updated and does not differentiate individual circuits and their characteristics.

The *Timing Side Channel* (cf. Section 3.1) refines the exit prediction based on the characteristics of a particular circuit. For simplification, we assume an adversary who can observe `nTor` handshakes from the entry position of a circuit to be established (e. g., by running a malicious entry relay). In this case, the adversary already knows the middle relay (since it is directly connected to it) and can observe the `nTor` handshakes between client and middle relay, and client and exit relay (without knowing about the exit's identity). The transmission time between middle and exit relay can be approximated by the difference between the two handshake round-trip times (as observed from the entry position):

$$RTT(m, x) = RTT(e, x) - RTT(e, m). \qquad (2)$$

From the transmission time between middle and exit relay, the adversary can determine probabilities for each exit relay candidate based on different propagation models. Those can be dependencies between transmission time and traveled distances or comparing the observed time with distributions of previously collected sets of transmissions times between individual relays, or between groups of relays. We introduce such specific propagation models and evaluate them as part of the exit prediction case study presented in Section 5.1.

Whereas we assumed that the adversary is located in the entry position here, it is also possible to transfer the concept to an adversary located between client and entry with access to the connection between them. In this case, the adversary must learn the middle relay's identity, which can principally be reached in a similar fashion. However, this two-step process adds more insecurity, since multiple relays may likely be in the middle position of the circuit, and likewise raise the efforts required, since the exit prediction must be conducted multiple times, i. e., for every likely middle relay. Exploiting the handshake timing side channel is thus independent of the attacker's exact position as long as there is access to some part of the entry connection.

4.3. Circuit Replacement

In this attack scenario, a client is forced to switch their primary guard and, therefore, to establish a new set of Tor circuits. Triggering the DoS mitigation in Tor in order to manipulate the circuit establishment process (cf. Section 3.3) leads to different routes taken between client and server, i. e. it comprises a routing attack on the application layer. We consider a scenario where clients use standard three-hop circuits to anonymously access regular web services that are publicly accessible. More specifically, we do not consider the use of onion services, which is more complex in terms of circuit establishment.

The concept described here helps an adversary who aims to observe the end-to-end connection of a specific client, but who has learned that parts of the connection are unreachable. Forcing the client to update their circuit set can increase the adversary's chances since the new relays (and therefore, the connections) may be in areas under their control, i. e. increasing chances for successful traffic analysis attacks.

4.3.1. Technical Attacker Capabilities. The guard rotation requires an adversary with the ability to spoof the IP address of a client, e. g., by using a TCP Man-in-the-Middle or by being located in the same NAT.

4.3.2. Concept. The adversary acts as follows to enforce a client to switch their primary guard as illustrated in Figure 5. First, they monitor the client's circuit establishments to determine its primary guard which is used for all circuits by default. Subsequently, the adversary's goal is to trigger the DoS mitigation of the client's primary guard. Therefore, the adversary impersonates the client's IP address towards the guard (cf. Fig. 5 (1)). When the adversary has successfully triggered the mitigation, e. g., by establishing a sufficient number of parallel connections, or exceeding the limits specified for other features (cf. Listing 1), the client cannot use its primary guard any more (2) and is forced to establish new circuits using a different guard in its guard set (3).

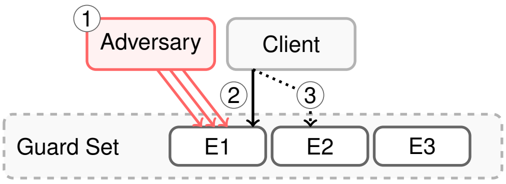

Figure 5. Exploit of DoS Mitigation. The adversary stresses the DoS mitigation in the first primary guard (1); the guard blocks the user's IP in response (2). The client cannot establish a connection anymore and continues with the second guard in the set (3).

This concept is not limited to a specific part of the connection an adversary is interested in. Specifically, it is not necessarily the entry connection that is under target. The idea behind this concept is that the client needs to generate an all new set of circuits, and therefore, resets the chances for an adversary who is unable to access a specific connection of interest. Thus, the guard rotation can be a means to gain access to a previously unobservable exit connection, as a means for end-to-end traffic analyses. The whole procedure described here remains stealthy, i. e., the guard cannot be accessed by the target client any more, but proceeds to operate normally for all other clients. From the client's perspective, the guard stops to operate and to accept new connections, but the client does not learn about the reason behind it, i. e., there is no reason for the client to assume being under threat.

4.4. Internal Denial-of-Service

Since all denial-of-service mitigation features in Tor only apply to unknown external IP addresses, the system still remains vulnerable to targeted denial-of-service attacks from inside. To achieve this, the adversary can transfer large amounts of traffic over Tor circuits such that the relays in this circuit reach their capacities. Such traffic must originate from Tor-internal IP addresses such as those used by regular relays. However, as Jansen et al. [21] show, denial-of-service can even be conducted from external addresses without triggering the mitigation, but depending on relay mitigation parameters, internal attacks from white-listed addresses always remain a fallback in the presence of tighter mitigation configurations.

Depending on the adversary's goals, denial-of-service affects Tor clients or the network as a whole at different dimensions, which we sketch with three examples at different scales. Directing a denial-of-service towards a single relay serves as an example to describe the general concept. We extend this by describing how scope and impact change when denial-of-service is conducted at larger scales, i. e., towards multiple relays at the same time or even targeting (large fractions of) the whole Tor infrastructure.

4.4.1. Technical Attacker Capabilities. For an internal denial-of-service, an adversary must either be able to spoof IP addresses of Tor relays or actually possess and control these address spaces. This is eligible for ISPs or AS providers, i. e., this variant is limited to very specific and powerful actors. Alternatively, a weaker adversary can set up their own set of relays and start conducting the attack as soon as the relay(s) have become part of the Tor consensus.

4.4.2. Single-Target DoS. Different strategies to realize denial-of-service against a single Tor relay have been shown to be feasible even from outside Tor without triggering DoS mitigation [21]. Strategies include establishing multiple circuits using the target relay and downloading large files to consume large amounts of bandwidth, or to include the target relay in a single 8-hop circuit multiple times. Therefore, we assume that such an attack scenario is realistic also from inside Tor with no DoS mitigation in effect.

Stressing a particular relay by generating large amounts of traffic renders the target relay unable to accept new connections for further circuits. This forces clients to include different relays in their circuits, thus, effectively re-routes client traffic. Target relays in such a scenario can be high-bandwidth guards or exits of interest that cannot directly be accessed by an adversary, e. g., one of the main guards or exits in a particular country. The adversary's goal is to increase chances for the substitute relay to be located in an area under adversarial control, likewise for entry or exit traffic being routed through the area.

While motivation and eventual outcome are quite similar to the guard rotation strategy in Section 4.3 when it is directed against a specific client, the denial-of-service attack is different in that it affects all clients using the target relay at the same time. When the adversary cannot conduct the rather stealthy guard rotation targeting a particular client, stressing the relay may still be a fallback option. However, since the strategy requires a more severe intervention and effectively tears down the relay, it can also be observed easier.

4.4.3. Multiple-Target DoS. Whereas the technical approach for the Multiple-Target DoS attack is essentially the same as for blocking a single relay, the main difference is that there are multiple targets simultaneously. Likewise, the goal of re-routing Tor traffic through areas that can be easier accessed by the adversary remains similar.

Groups of targets can be, e. g., all relays in a specific (unreachable) AS, or all relays of a particular country. In Section 5.3, we take a closer look into the feasibility of such an attack scenario given the nature of the real Tor infrastructure.

4.4.4. Network DoS. Further extending the denial-of-service to a large fraction of all relays not only drastically reduces the choice of relays that clients have to construct their circuits (and therefore, their overall anonymity set within Tor). Likewise, denial-of-service at large scale also affects the stability and reliability of the whole Tor infrastructure. Distributing the steady Tor traffic across a significantly smaller fraction of relays that, in turn, may not be able to handle the increased amounts of traffic can even amplify the attack, eventually cascading across the whole Tor infrastructure. The technical approach still remains the same – the adversary generates large amounts of traffic that exceed bandwidth capacities of target relays. However, conducting the attack at larger scale simply requires more resources.

5. Case Studies

We present three empirical simulations as case studies for the Exit Prediction, Circuit Replacement, and Multiple-Target Denial-of-Service attacks. For each case study, we introduce the specific scenario, i. e., the empirical data the simulation relies on, explain how we evaluate the attack performance in this scenario, and present the results.

5.1. Exit Prediction

Combining `nTor` handshake timing data with relay distribution information allows assigning relays a probability for being used in a particular Tor circuit. In this section, we conduct a general empirical evaluation to analyze the feasibility of predicting the exit node of a Tor circuit in a practical scenario. In the next step, we evaluate to what extent the results of an exit prediction serve as a stepping-stone for end-to-end confirmation attacks. To this end, we analyze how the exit prediction reduces the otherwise immense overhead of processing the recorded traffic of multiple connection endpoints within Tor.

5.1.1. Evaluation Data Set. To protect the security and privacy of real-world Tor users, we initially gather an empirical data set of Tor circuits that enables us to later *simulate* the exit prediction. Over one week, we record handshake timings with four different remote servers in New York, Amsterdam, London, and Frankfurt that act as Tor clients; we record two different types of circuits. First, *standard circuits* consist of relays that a client picks, i. e., they resemble the original selection criteria of Tor. Second, we extend the set of standard circuits by artificial *random circuits* that provide us with diverse transmission characteristics. Overall, we measure roughly 84,500 standard and 172,500 random circuits. We use this empirical data set for two main purposes.

Propagation Model. As we predict possible exit locations from the monitored times of the circuit establishment handshakes, we depend on a realistic model of Tor's transmission characteristics. Such a model allows us to compare the measured times with general characteristics like propagation times and their relation to the traveled distance. Therefore, we use the empirical data set to derive a propagation model that we later use to estimate the target locations of exits. We use the generated data of roughly 257,000 handshakes to aggregate distributions of transmission times between pairs of countries the relays under consideration are located in. We generate probability density functions for transmission times between all pairs of countries. That is, we can determine a probability for a specific transmission time to have occurred in transmission between two countries. For the evaluation, we use 10-fold cross-validation, i. e., we compute 10 sets of empirical time distributions, each one leaving out 10 % of the *standard* circuits.

Exit Prediction Simulation. The monitored circuits serve as a test set for the exit prediction simulation. We randomly pick 10,000 standard circuits and predict the exit relay for each of them. Since we know the correct relays, we can use this information to measure the quality of a prediction. When we predict the exit of a standard circuit

TABLE 1. Evaluation of exit prediction. Median relative ranks of the true exit across all predictions, by exit country.

	DE	US	FR	GB	CH	NL	AT	SE	RO	CA
COMB	4 %	12 %	7 %	9 %	10 %	8 %	1 %	6 %	11 %	18 %
TIME	10 %	25 %	13 %	15 %	18 %	17 %	7 %	16 %	25 %	18 %
BW	11 %	21 %	16 %	23 %	22 %	22 %	1 %	13 %	18 %	35 %
RAND	49 %	50 %	50 %	51 %	53 %	50 %	49 %	50 %	48 %	52 %

with the propagation model described above, we ensure to always use the model instance that the circuit under consideration was not included in (*cross validation*).

5.1.2. Evaluation Metrics and Assumptions. For the exit prediction, we generate and compare four different probabilistic relay rankings described here. First, we consider a bandwidth-based (**BW**) prediction that simply assigns a probability $P_{bw}(x)$ to each exit x depending on its bandwidth fraction. Second, we consider a (**TIME**) prediction based on `nTor` handshake timing information. Given that the location (country) of the middle relay is already known (cf. Section 4.2), we can determine a probability $P_{time}(x)$ for each exit relay x by considering its country and look up the likelihood for timing value in the distribution for the corresponding country pair (middle, exit) in the propagation model described above. We also consider a combined (**COMB**) prediction that takes into account both probabilities in combination. Since both observations are independent of each other, we determine the combined probability $P_{comb}(x)$ as follows:

$$P_{comb}(x) = P_{bw}(x) \cdot P_{time}(x) \tag{3}$$

Solely for reference, we also provide results for a prediction with all relays ordered randomly (**RAND**). However, ranking all relays in random order is not a realistic strategy. Since relays with higher bandwidth are more likely to be picked for a circuit, we consider a bandwidth-based ranking the baseline strategy for a strategic attacker.

When evaluating the accuracy of the four different predictions, we aggregate the relays by country. This aggregation results in sets of relays, each of which a potential nation-state adversary is able to observe. We focus on nation-state adversaries, as each country has its own concept to treat Tor traffic. This results in individual jurisdictions where all traffic through the country experiences the same "treatment", e. g., legal regulations that consider Tor traffic as suspicious will allow the monitoring of transmissions. Considering a nation-state adversary allows us to predict the consequences for potentially malicious key countries of Tor's infrastructure.

For each nation-state, we consider the median relative predicted rank across all cases in which the circuit's true exit was located in the respective country. As an example, for all circuits whose exit is located in Germany (DE), we denote the median relative rank of the true exits across all predictions for these circuits.

For each country, we further evaluate how the outcome of an exit prediction can help an adversary in conducting end-to-end confirmation attacks more strategically. To this end, we evaluate how the exit prediction is a stepping-stone to successful and resource-efficient traffic confirmation as the adversary only attacks a specific fraction of the exit prediction ranking. To obtain these results, we simulated the exit prediction for 10,000 circuits randomly picked from our evaluation data set.

5.1.3. Results. Table 1 presents the exit prediction performance. The results show the median relative ranks of the true exit across all predictions sorted by exit country. The combined prediction (COMB) achieves the most accurate results across all countries; we provide the results of a randomized prediction (RAND) for reference. We focus on the top 10 countries w. r. t. to their total exit bandwidth. A lower value indicates a better ranking of the actual exit in the prediction, thus, a higher prediction accuracy. Due to the skewed distribution of resources within the Tor infrastructure, the results for the prediction models vary across different countries.

General Performance. We now compare the results in the US and Germany (DE), two essential countries for the Tor infrastructure in the number of relays and the bandwidth they provide. In the median case, a relay located in Germany is ranked in the top 10 % with the time-based exit prediction and in the top 11 % using the bandwidth-based prediction. When combining the two approaches, any relay located in Germany is ranked in the top 4 % of all relays in the median case. The US appears to be an exception, with the prediction performing worse than for other countries, particularly w. r. t. the time-based prediction. We attribute this to different geographical circumstances, e. g., significantly longer transmission distances than across the other countries located in Europe. However, combining the timing and bandwidth ranking still provides a median ranking within the top 12 % of relays. For the remaining countries, we see similar performances for both individual prediction metrics, with the timing-based prediction performing slightly better. Combining the two approaches improves the performances for relays located in all countries under consideration.

Stepping-Stone. A nation-state adversary operating in a particular country can use the outcome of the exit prediction to act more strategically and reduce its efforts for subsequent traffic-analysis attacks targeting particular Tor circuits. We now analyze how successful one exemplary adversary (US) can be in end-to-end confirmations and how much exit traffic they need to monitor when they only monitor their relays ranked above a specific threshold in the exit prediction ranking. For reference, an adversary without assumptions about the actual exit (*baseline*) would always monitor *all* of their relays and analyze traffic in decreasing order of relay bandwidth until their end-to-end correlation has been successful (i. e., implicitly follow the bandwidth-based ranking), since relays with higher bandwidth generally have a higher likelihood to be picked for a circuit.

Figure 6 illustrates (a) what fractions of accessible traffic the adversary can observe (i. e., their expected suc-

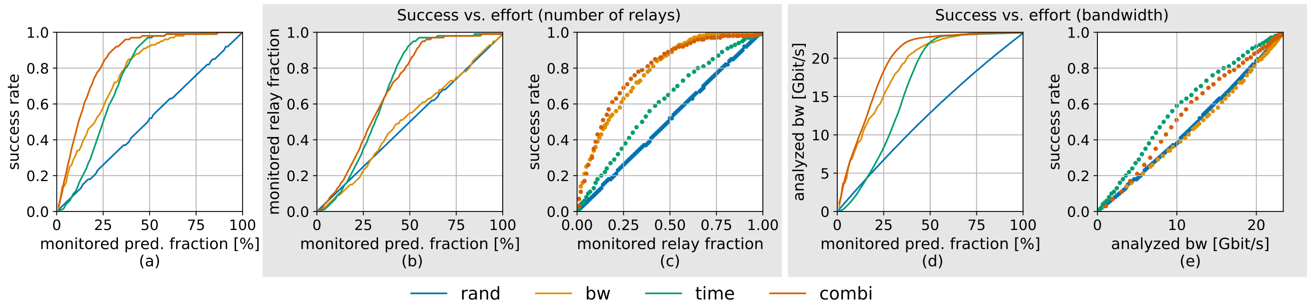

Figure 6. Detailed Exit Prediction Performance Evaluation for the US adversary. Depending on the fraction of relays in a prediction that an adversary monitors, (a) shows the relative success rate, i. e., what fractions of accessible traffic an adversary can observe when monitoring relays within a specific fraction of the prediction. The next two blocks (each highlighted in gray) compare success and adversarial effort in two steps. While (b) shows the fraction of relays under adversarial control within a specific prediction fraction, (c) combines (a) and (b) by showing the success rate relative to the monitored relay fraction. Likewise, (d) shows the bandwidth to be analyzed when monitoring a specific fraction of the prediction, and (e) compares the analyzed bandwidth with success rates. Results for more adversaries (top 10 countries in exit bandwidth) can be found in the appendix.

cess rates, y-axis) when they only monitor relays ranked within a specific fraction of the prediction (x-axis), i. e., above a specific *threshold rank*. While (a) directly connects the success rate to the outcome of a prediction, it does not consider the adversarial effort required to monitor all relays above the threshold in the prediction. Accordingly, (b) shows the fraction of relays the adversary monitors, i. e., what fraction of their relays is ranked above the threshold, and (c) combines (a) and (b) showing how the success rate depends on the monitored fraction of relays. In the same way, (d) and (e) connect monitored prediction fractions to analyzed relay bandwidths and to success rates in two steps.

The bandwidth that needs to be analyzed (depending on the monitored ranking fraction) is determined as follows: The adversary analyzes all traffic streams following the order in which they are ranked in the prediction. In case the adversary is *not* able to access the exit traffic that relates to the entry point of the connection, i. e., both parts of the end-to-end confirmation, the traffic of all relays above the threshold rank needs to be analyzed (without success). In case the adversary *is* able to access exit traffic, only the traffic streams of the actual relay and all relays ranked *higher* in the prediction need to be analyzed. There is no need to continue analyzing the rest of the monitored traffic streams as soon as the end-to-end confirmation has been successful. We assume that the underlying analysis technique is able to reliably distinguish between related and unrelated streams, i. e., a correct match of related traces always leads to a clear result.

As we can see in Figure 6 (e), the timing-based and combined prediction achieve higher success rates per analyzed bandwidth than following the bandwidth-based relay ranking of the baseline adversary.

An adversary who analyzes 10 Gbit/s of exit traffic achieves a success rate of 39 % when analyzing their relays in decreasing order of bandwidth. When monitoring the nTor handshake and ranking relays based on these timings (or in combination with bandwidth information, respectively), the adversary can achieve a success rate of 59 % (timing-based ranking) when analyzing the same amount of traffic.

Table 2 corresponds to the US results in Figure 6 (e) and lists AUC (Area Under the Curve) values for the success rates, where larger values indicate a better performance. We use the AUC to summarize the overall performance of a nation-state. As we see, the bandwidth-based prediction (**bw**) achieves a similar performance (in terms of success per analyzed exit bandwidth) as the randomized exit prediction (**rand**) in all countries. The time-based prediction (**time**) and the combination of timings and bandwidth (**combi**) achieve higher performances across all countries, e. g., the timing-based prediction performance in the US is 27 % higher than the bandwidth-based prediction. Please note that these numbers can only be compared between predictions *within* the same country due to the different exit bandwidth amounts across different countries.

TABLE 2. EXIT PREDICTION PERFORMANCES IN TERMS OF SUCCESS RATES PER ANALYZED EXIT BANDWIDTH.

Country	**rand**	**bw**	**time**	**combi**
US	11.02	10.81	13.74	12.61
DE	2.33	2.31	2.86	2.64
FR	1.48	1.41	1.48	1.53
GB	2.34	2.14	2.51	2.29
CH	3.52	3.48	3.94	3.84
NL	3.30	3.30	3.95	3.66
AT	4.91	4.23	6.11	4.72
SE	1.11	1.06	1.29	1.26
RO	1.35	1.31	1.58	1.54
CA	1.44	1.45	1.91	1.92

Our results imply that strategically analyzing exit traffic based on observed handshake timings can actually increase the adversary's success for end-to-end confirmation attacks, or reduce their required efforts, accordingly. That is, the adversary has a choice in the selection of the ranking strategy and, consequently, the trade-off between accuracy and analysis overhead.

5.2. Circuit Replacement

We now evaluate the impact of the circuit replacement attack by simulating how nation-state adversaries can improve their chances of observing Tor exit traffic by enforcing the guard rotation directed against a specific client.

TABLE 3. IMPROVEMENT [%] THROUGH CIRCUIT REPLACEMENT

DE	US	FR	GB	CH	NL	AT	SE	RO	CA
5	33	4	6	8	7	15	3	3	5

5.2.1. Evaluation Data Set. We use our experimental setup as described in Section 5.1 to generate test sets of Tor circuits as they are pre-built in a client-side Tor instance. For each test set, we consider a single guard node in the entry position of all circuits and add up to 1,000 circuits to the test set. We only take into account guard relays for which we have a minimum of 50 circuits with that relay in the entry position, resulting in 37 different guards to consider.

5.2.2. Evaluation Metrics and Assumptions. We consider the fraction of circuits with the exit node located in an adversarial area within each test set, representing a client's updated guard use after a circuit replacement attack has been conducted. Given that an adversary cannot access the exit traffic in a particular circuit, these numbers represent the chance of accessing the traffic after the attack has been conducted.

We define adversarial areas on a per-country basis, focusing on the top 10 countries w. r. t. to their total exit bandwidth.

We assume that clients behave regularly and only use circuits with their primary guard in the entry position and only switch to circuits with their secondary guard when the primary guard becomes unavailable. Furthermore, a client has a usage profile that puts similar loads on all available circuits, i. e., we consider all circuits with the same entry guard equally.

We now provide our results of the application layer routing attack simulation on a per-country basis.

5.2.3. Results. The circuit replacement can be used for additional attempts to gain access to both ends of the circuit of a specific user. That said, an adversary conducts the attack in cases where *no* access to the exit relay or traffic is given. Table 3 summarizes the achievable *improvements* for nation-state adversaries who can access the traffic of all relays within their area. The numbers represent the average fractions of circuits with exits in the respective country after the circuit replacement. One example of a substantial improvement is the US. On average, one third of exits are located in their area after conducting the circuit replacement. The improvements we report in Table 3 roughly match the bandwidth fractions of all exit relays located in the respective countries as they appeared in the consensus used for the simulation. This is a plausible outcome since relays to be included in a circuit are mainly selected based on their bandwidth. However, these numbers are subject to constant change, depending on the evolution of the Tor infrastructure and changes in bandwidth distributions. As of March 2021, the bandwidth fraction of exit relays in the US has dropped to 20 %; the fraction of exits in DE has increased to 31 % with (presumably) equal consequences for the attack success.

5.3. Multiple-Target DoS

We evaluate the impact of internal denial-of-service directed at multiple target relays. We consider the bandwidth cost required to stress the relays under target and how the attack can improve the adversary's chances in a traffic analysis attack scenario.

5.3.1. Evaluation Data Set. Our evaluation is based on a single Tor consensus (as of 23 October 2020) supplemented with location data and AS information retrieved from an IP geolocation service. The consensus contains 6,735 relays in total, 3,717 of which have a guard flag and 1,427 can be used as exit nodes.

5.3.2. Evaluation Metrics and Assumptions. In order to evaluate the practicality of a multi-target denial-of-service attack in Tor, we assume that a relay can be effectively stressed by generating the amount of traffic that corresponds to its assumed link capacity. We refer to this as *DoS bandwidth*. Following the approach of Jansen et al. [21], we consider the relay bandwidth as advertised in the consensus and estimate its link capacity as the next higher value in a set of fixed bandwidth classes (1M, 10M, 100M, 200M, 500M, 1G, 10G [bit/s]).

We consider DoS bandwidths on a per-country basis, resembling a scenario in which a nation-state adversary with access to all relays in a particular country aims to increase their chances for accessing Tor traffic by targetedly disabling relays in areas out of reach.

Finally, we estimate the cost to perform such an attack by taking into account the amount of traffic that is required to stress a relay with a given link bandwidth over a specific period of time.

5.3.3. Results. The DoS bandwidth required to stress all relays varies across different countries, depending on how much bandwidth relays in these countries provide and how the bandwidths are distributed across all relays in a country. In Table 4, we present required DoS bandwidths for stressing all guards and exits in the top 5 countries w. r. t. to the provided bandwidth.

We see that guard bandwidths are higher than exit bandwidths across all countries, therefore also requiring higher DoS bandwidths when targeting guards. We also observe that higher total bandwidths per country do not directly translate into higher DoS bandwidths. When considering the guard bandwidth per country, the US, GB, and

TABLE 4. RELAY BANDWIDTH VS. REQUIRED DOS BANDWIDTH.

	Country	Relays	Total BW [Gbit/s]	Fraction	DoS BW [Gbit/s]
Guards	**DE**	863	208.92	35.6 %	1,228
	FR	561	105.15	17.9 %	420
	US	615	46.65	7.9 %	87
	GB	175	45.25	7.7 %	197
	NL	227	44.70	7.6 %	223
Exits	**DE**	289	78.98	35.7 %	305
	US	355	30.38	13.7 %	50
	FR	126	27.42	12.4 %	60
	GB	103	26.27	11.9 %	53
	NL	63	11.97	5.4 %	52

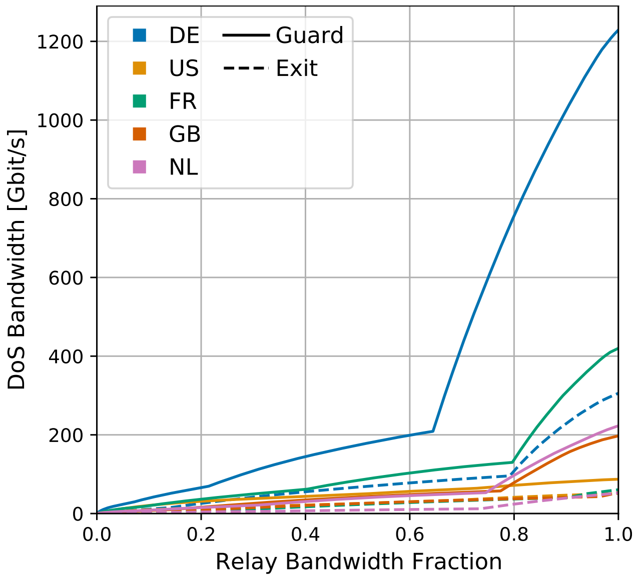

Figure 7. DoS bandwidth cost for bandwidth fractions of guards and exits per country.

NL provide roughly 8 % of the overall guard bandwidth each but require different amounts of DoS bandwidths to be stressed. These differences can also be seen when considering the required DoS bandwidths per individual country for Guards and Exits (cf. Figure 7). For example, stressing 80 % of the exit bandwidth in DE requires roughly 100 Gbit/s of DoS bandwidth, whereas stressing the remaining 20 % additionally requires 200 Gbit/s of DoS bandwidth, i. e., twice as much added on top. The reason for this is the different distributions of bandwidths within these countries. For all European countries (DE, FR, GB, NL), we see that there is a small fraction of relays individually contributing high bandwidths of up to 1,000 Mbit/s, and in some cases even above. Due to the assumption of considerably higher link bandwidths in this case, a small number of high-bandwidth relays largely influences the overall required DoS bandwidth in these countries. In contrast, such high-capacity relays are not present in the US, which implies that a single relay can add a maximum of 500 Mbit/s to the total required DoS bandwidth for this country. We provide an overview of the bandwidth distributions for all guards and exits in the top 5 countries in the appendix (cf. Figure 10).

For estimating the practicality of a multi-target denial-of-service attack we consider an adversary who aims to increase their chances for successfully conducting a traffic analysis attack. We consider a nation-state adversary who is able to access all relays located in Germany. Essentially, this scenario resembles a simple statistical calculation. Due to the fraction of 35.7 % of relay bandwidth in DE, the adversary is already in a comfortable situation in being able to access more than one third of exit traffic initially. However, targetedly disabling all exit relays in the other four countries we consider, requires roughly 215 Gbit/s of DoS bandwidth (cf. Table 4). In return, 44 % of Tor's overall exit bandwidth is rendered unavailable. Within the set of remaining exit relays, the bandwidth fraction of relays in Germany increases to 64 %, which means that the chances of exit traffic using a relay in the adversarial area have almost doubled.

TABLE 5. Overview of traffic cost for the DoS attack using common server providers.

Provider	Cost/GB	Cost/500M/hour
Azure	$0.09	$20.25
AWS	$0.15	$33.75
Google Cloud	$0.12	$27.00

Cost Estimation The cost to conduct the denial-of-service attack is mainly driven by the cost to produce large amounts of traffic. To estimate the cost, we take into account the amount of traffic that is required to stress a relay's link bandwidth for one hour. This time-span is sufficient for a targeted attack over a limited period of time. Fully utilizing a link bandwidth of 500 Mbit/s for one hour requires an adversary to generate 225 GB of traffic. This amount seems appropriate for our estimation since, e. g., every relay in the US has a lower assumed link bandwidth. Table 5 provides an overview of the corresponding cost using a few large server providers. This means that disabling one relay with an assumed link bandwidth of 500 Mbit for one hour can be purchased for around $ 20. The cost for 215 Gbit/s of DoS bandwidth for disabling all exit relays in US, FR, GB, and NL for one hour (i. e., 96.75 TB of traffic) sum up to $ 8,700. These estimations do not consider the cost of running a Tor relay to conduct these attacks from inside the Tor network. However, since the cost for running a standard server instance (which can be used to run the relay) can be kept well below $ 10 per month and one host has a link bandwidth of 10 Gbit/s (among the major server providers, which is sufficient for attacking 20 smaller targets in parallel), these costs seem negligible.

5.4. Ethics Considerations

During our measurements, we have taken great care to adhere to the principles of ethical research and did our best not to pose any threats to real Tor users or parts of the Tor infrastructure.

In the experiment to determine the cryptographic overhead of the `nTor` handshake times (Section 3.1), we ran a publicly accessible relay for general use in Tor. We did not collect any data other than timestamps and did not harm the anonymity of Tor users connecting to our relay at any time.

In order to validate the DoS mitigation behavior as a means to enforce Guard Rotation (Section 3.3), we also ran a publicly accessible Tor relay. In order to minimize its chances for being picked by other clients, we limited the offered bandwidth rate to make it one of the less prominent relays in the consensus. At no point in time, we monitor or interfere with connections from other users.

During the data collection for the exit prediction case study (Section 5.1), we established roughly 257,000 circuits but did not actively create payload traffic utilizing the relays involved. In comparison to Tor's daily load, the amount of traffic we created is negligible and did not impair the use of the system.

6. Discussion

The threat vectors introduced in this work serve as a stepping stone for follow-up traffic analysis attacks. As they exploit characteristics of core and defensive features within Tor, these threat vectors are hard to counter and cannot simply be removed through an update of Tor. In the following, we discuss alternative directions that can help to limit the success of the stepping-stone attacks of this work.

6.1. Impact of DoS Attacks

Because of its voluntarily operated infrastructure, DoS attacks against Tor can be conducted easily. Prior work demonstrates how adversaries can disable critical nodes in the network targetedly [20], [21]. The DoS mitigation features that have been introduced in response aim to recognize and block excessive circuits, connections, and cells. However, we saw that they also *introduce a new threat vector* (cf. Section 3.3). Besides this guard rotation, the current DoS mitigation setup further allows continuing DoS attacks against relays from within the Tor infrastructure.

Improving the DoS Mitigation. The guard rotation is a reminder of the elaborate design and deployment of new defensive concepts for a live system. Other than the `nTor` handshake procedure, which is a core requirement for the onion encryption of Tor traffic, the DoS mitigation is a defensive mechanism introduced in response to a specific type of attack. While we see that it protects against DoS attacks targeted at clients, its simple concept does not manage to protect relays, nor does it avoid being exploited for other types of attacks against clients. We recommend updating the DoS mitigation in a way that blocks DoS attacks against relays without restricting maintenance traffic (instead of entirely skipping the detection for known IP addresses) and to add a client notification that allows recognizing an exploit of the client DoS mitigation.

6.2. Protecting against Exit Prediction

In contrast, the `nTor` handshake is a core mechanism to enable onion encryption, and every circuit build-up depends on it. However, the handshake messages are not obfuscated, and their end-to-end timing allows us to derive the round trip times between single hops of the circuit.

6.2.1. Timing Obfuscation. As transmissions through the Internet are often affected by asymmetric routing or congestion, the attack already uses a noisy information source. For a countermeasure, we can use these effects and add further random delays to messages of the handshake. Unlike mixing an entire connection, which introduces an unacceptable overhead, delays in the handshakes are limited to only a few messages and keep the overhead to a minimum. Another option is the use of pluggable transports [6], [29], [49] that obfuscate Tor entry traffic.

Table 6 lists the performances of the timing-based exit prediction (AUC of success rates per analyzed exit bandwidth, cf. Table 2) for the top 10 countries in terms of exit bandwidth. We compare the performances of the time-based prediction based on original handshake timings and randomly delayed handshake timings. The left column (*No Delay*) corresponds to the *time* column in Table 2. The delay amounts in the other columns denote the maximum delay added to each handshake; the individual delay for each handshake was drawn uniformly at random between 0 and the maximum delay.

TABLE 6. Performance of the timing-based exit prediction with and without timing obfuscation.

Country	No Delay	Delay (0.1 s)		Delay (0.2 s)	
US	13.74	10.37	-25 %	10.93	-20 %
DE	2.86	2.7	-6 %	2.65	-7 %
FR	1.48	1.21	-18 %	1.19	-20 %
GB	2.51	2.26	-10 %	2.79	11 %
CH	3.94	3.62	-8 %	3.47	-12 %
NL	3.95	3.57	-10 %	4.01	2 %
AT	6.11	5.68	-7 %	5.29	-13 %
SE	1.29	1.16	-10 %	0.94	-27 %
RO	1.58	1.34	-15 %	1.13	-28 %
CA	1.91	1.74	-9 %	1.7	-11 %

The results imply that performances of the timing-based prediction can be reduced by up to 21 % (US) when we introduce random delays with a maximum of 0.1 s in the handshakes. This amount of time seems sufficient since we do not observe any added gain when increasing the maximum delay to 0.2 s. With random delays applied, the performance of the timing-based prediction is similar to performances of random and bandwidth-based predictions (cf. Table 2).

In conclusion, delaying the `nTor` handshake may be sufficient to prevent from exploiting the timing information at the expense of acceptable timing overhead; however, prediction can still be conducted based on relay bandwidth fractions.

6.2.2. Randomized Relay Selection. Adding randomness to the relay selection process may be helpful to hamper adversarial strategies. However, it must be carefully evaluated to what extent the trade-off between security and performance can be further shifted towards security while paying with additional latency.

We compare the circuit bandwidths (i. e., the sum of bandwidths of the three relays in the circuit) of standard Tor circuits and circuits with randomly selected relays. Figure 8 shows the distributions of circuit bandwidths for 10,000 circuits of each type. The average circuit bandwidth of 1,010 Mbit/s for weighted circuits (median: 955 Mbit/s) decreases to 350 Mbit/s (median: 270 Mbit/s) for random circuits, which comprises an average reduction down to one third.

Besides the connected performance reduction for Tor users, this issue may also cause problems for relay providers. With no adequate load-balancing mechanisms in place (i. e., bandwidth-based selection), especially lower bandwidth relays are used more frequently, even when their original bandwidth capacities are exceeded. Eventually, this renders them unavailable. We leave a more detailed evaluation of these issues an open task for future research.

6.2.3. Uniform Infrastructure. The skewed distribution of relays makes it difficult to avoid certain combinations of nodes. For example, avoiding a nation-state adversary in

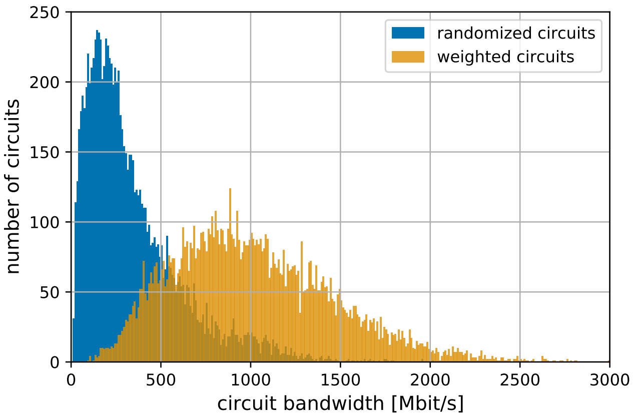

Figure 8. Bandwidth comparison for standard Tor circuits vs. circuits with randomly selected relays.

one of the main infrastructure-providing countries leads to severe performance impairments for a user. Prior work addresses such circumvention of untrusted areas and comes to the conclusion that geographical avoidance is possible from a technical perspective but infeasible for specific countries in the Tor infrastructure [26]. A more uniform distribution of nodes, and bandwidth in particular, improves this situation. However, we emphasize that the Tor network consists of a voluntarily operated infrastructure without any central management of relay locations.

7. Related Work

We describe related work on routing attacks, Tor path selection, and traffic analysis attacks.

Routing Attacks and Defenses. The Border Gateway Protocol (BGP) defines service agreements between Autonomous Systems (AS), i. e., it provides and manages the collection of rules for routing traffic between different ASes. Attacks on the BGP influence how traffic gets routed between the source and destination of a connection and can help the adversary to gain access to transmissions. The idea of performing DoS attacks to re-route traffic of anonymous communication networks through areas under adversarial control has first been brought up by Borisov et al. [7]. Other examples of routing attacks are BGP hijacks [33], [42] that blackhole traffic, which allows observing data but results in a dropped connection. BGP interceptions [3] achieve similar results, but manage to keep connections alive and are less detectable. In contrast to these attacks, our concept for an application-layer routing attack exploits one of Tor's security features instead of directly manipulating the routing behavior.

Defenses against routing attacks include trust-based concepts for anonymity systems [24], where a trust model of the Internet helps to identify and avoid potentially dangerous areas. In their work, the authors introduce an algorithm for the path selection in onion routing systems to protect the anonymity of users against adversaries in control of large fractions of the network. Other approaches consider data-plane defenses against specific routing attacks on Tor that use control-plane manipulations [43]. One down-side of such defenses is the assumed attacker model, as colluding AS-level adversaries and nation-states have access to a majority of the network.

Path Selection in Tor. Obfuscating the path of a circuit to an external server is essential to provide anonymity within Tor, and the characteristics of its path selection procedures have been shown to affect the anonymity of Tor clients [2], [12], [25], [46]. Shizari et al. [39] present a classification of anonymous routing protocols, and find node selection being a common strategy to improve anonymity in onion routing protocols. Panchenko et al. [35] propose a path selection method with improved performance and anonymity by considering the current relative load of Tor nodes. There are proposals to select Tor circuits based on their performance, picking less congested nodes [4], [48]. With respect to better anonymity, there are proposals for location-aware path selection to avoid adversarial ASes [1], [5], [22]. In this context, Wan et al. [47] show how strategical guard placement can overcome the defenses of such approaches. Snader and Borisov [40] propose to let users choose between performance and anonymity. The higher performance is weighed by users, the higher relay bandwidth is weighed in composing a circuit. Cangialosi et al. [8] have shown that Tor circuits longer than three hops can achieve lower end-to-end latencies. Sophisticated path selection algorithms can help to avoid untrusted areas, which reduces the overall probability of being targeted in an attack. However, large-scale adversaries with access to a majority of network resources can hardly be circumvented and, thus, limit the defensive effects of secure path selection.

Traffic Analysis Attacks. Website fingerprinting, i. e., classifying encrypted traffic stream data based on patterns of previously collected data, enables an adversary to observe the websites or services a user connects to, without actually accessing the encrypted traffic. In this context, Overdorf et al. [34] analyzed features of fingerprinting classifiers to evaluate the uniqueness of onion services. Hopper et al. [15] present a linkability attack, exploiting latency information to predict whether two connections use the same Tor circuit. Kwon et al. [28] showed that Tor circuits connecting to onion services could be distinguished from regular Tor circuit traffic, which is an information leak w. r. t. users accessing Tor hidden services. Jansen et al. [19] developed a framework that identifies onion services solely from middle traffic. Jaggard et al. [18] demand research to put focus on adversaries targeting specific users, since this type of adversary is more realistic. The two attacks we introduce in this work are a stepping stone for follow-up traffic analysis attacks targeting particular users and help to manipulate and predict candidates for potentially related connection endpoints.

8. Conclusion

In this paper, we introduced a set of stepping-stone attacks exploiting previously under-studied threat vectors within core mechanisms in Tor. These attacks can facilitate traffic analysis attacks in different ways. The result of an exit prediction hints an adversary towards likely candidates for the exit relay in a Tor circuit. This can either reduce their efforts required for traffic analyses by enabling them to act more targeted, or keep them from attempting an attack that will most likely be unsuccessful. In contrast, active adversary intervention in enforcing the targeted and stealthy guard rotation can provide additional attempts for traffic analyses that turned out to be impossible before. Similarly, targeted stressing sets of Tor relays can be used to enforce routing through adversarial areas, additionally affecting the reliability and stability of the Tor infrastructure.

References

[1] Masoud Akhoondi, Curtis Yu, and Harsha V. Madhyastha. LASTor: A Low-Latency AS-Aware Tor Client. In *IEEE Symposium on Security and Privacy*, SP '12, pages 476–490, San Francisco, CA, USA, May 2012. IEEE.

[2] Michael Backes, Aniket Kate, Sebastian Meiser, and Esfandiar Mohammadi. (Nothing else) MATor(s): Monitoring the Anonymity of Tor's Path Selection. In *ACM Conference on Computer and Communications Security*, CCS '14, pages 513–524, Scottsdale, AZ, USA, November 2014. ACM.

[3] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A Study of Prefix Hijacking and Interception in the Internet. *ACM SIGCOMM Computer Communication Review*, August 2007.

[4] Armon Barton, Mohsen Imani, Jiang Ming, and Matthew Wright. Towards Predicting Efficient and Anonymous Tor Circuits. In *USENIX Security Symposium*, USENIX '18, pages 429–444, Baltimore, MD, USA, August 2018. USENIX Association.

[5] Armon Barton and Matthew Wright. DeNASA: Destination-Naive AS-Awareness in Anonymous Communications. In *Proceedings on Privacy Enhancing Technologies Symposium*, PoPETS '16, pages 356–372. De Gruyter, October 2016.

[6] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange. Elligator: Elliptic-Curve Points Indistinguishable from Uniform Random Strings. In *ACM Conference on Computer and Communications Security*, CCS '13, pages 967–980, Berlin, Germany, November 2013. ACM.

[7] Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. Denial of Service or Denial of Security? In *ACM Conference on Computer and Communications Security*, CCS '07, pages 92–102, Alexandria, Virginia, USA, October 2007. ACM.

[8] Frank Cangialosi, Dave Levin, and Neil Spring. Ting: Measuring and Exploiting Latencies Between All Tor Nodes. In *ACM SIGCOMM Conference on Internet Measurement*, IMC '15, pages 289–302, Tokyo, Japan, October 2015. ACM.

[9] Heyning Cheng and Ron Avnur. Traffic Analysis of SSL Encrypted Web Browsing, 1998.

[10] George Danezis. The traffic analysis of continuous-time mixes. In *Workshop on Privacy Enhancing Technologies*, PET '04, pages 35–50, Toronto, Canada, May 2004. Springer.

[11] 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 *Workshop on Privacy in the Electronic Society*, WPES '12, Raleigh, NC, USA, October 2012. ACM.

[12] Nathan S Evans, Roger Dingledine, and Christian Grothoff. A Practical Congestion Attack on Tor Using Long Paths. In *USENIX Security Symposium*, USENIX '09, pages 33–50, San Diego, CA, USA, June 2009. USENIX Association.

[13] John Geddes, Rob Jansen, and Nicholas Hopper. How Low Can You Go: Balancing Performance with Anonymity in Tor. In *Privacy Enhancing Technologies Symposium*, PETS '13, pages 164–184, Bloomington, IN, USA, July 2013. Springer.

[14] Jamie Hayes and George Danezis. Guard Sets for Onion Routing. In *Proceedings on Privacy Enhancing Technologies Symposium*, PoPETS '15, pages 65–80. De Gruyter, 2015.

[15] Nicholas Hopper, Eugene Y Vasserman, and Eric Chan-Tin. How Much Anonymity Does Network Latency Leak? *Transactions on Information and System Security (TISSEC)*, 13(2):1–28, 2010.

[16] Amir Houmansadr and Nikita Borisov. The need for Flow Fingerprints to Link Correlated Network Flows. In *Privacy Enhancing Technologies Symposium*, PETS '13, pages 205–224, Bloomington, IN, USA, July 2013. Springer.

[17] Amir Houmansadr, Negar Kiyavash, and Nikita Borisov. RAINBOW: A Robust And Invisible Non-Blind Watermark for Network Flows. In *Network and Distributed System Security Symposium*, NDSS '09, San Diego, CA, USA, February 2009. The Internet Society.

[18] Aaron D Jaggard and Paul Syverson. Onions in the Crosshairs: When The Man Really is out to get you. In *Workshop on Privacy in the Electronic Society*, WPES '17, pages 141–151, Dallas, TX, USA, 2017. ACM.

[19] Rob Jansen, Marc Juarez, Rafa Galvez, Tariq Elahi, and Claudia Diaz. Inside Job: Applying Traffic Analysis to MeasureTor from Within. In *Network and Distributed System Security Symposium*, NDSS '18, San Diego, CA, USA, February 2018. The Internet Society.

[20] Rob Jansen, Florian Tschorsch, Aaron Johnson, and Björn Scheuermann. The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network. In *Network and Distributed System Security Symposium*, NDSS '14, San Diego, CA, USA, February 2014. The Internet Society.

[21] Rob Jansen, Tavish Vaidya, and Micah Sherr. Point Break: A Study of Bandwidth Denial-of-Service Attacks against Tor. In *USENIX Security Symposium*, USENIX '19, pages 1823–1840, Santa Clara, CA, USA, August 2019. USENIX Association.

[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. In *Network and Distributed System Security Symposium*, NDSS '17, San Diego, CA, USA, February 2017. The Internet Society.

[23] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries. In *ACM Conference on Computer and Communications Security*, CCS '13, pages 337–348, Berlin, Germany, November 2013. ACM.

[24] Aaron M Johnson, Paul Syverson, Roger Dingledine, and Nick Mathewson. Trust-Based Anonymous Communication: Adversary Models and Routing Algorithms. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 215–226, Chicago, IL, USA, October 2011. ACM.

[25] Joshua Juen, Aaron Johnson, Anupam Das, Nikita Borisov, and Matthew Caesar. Defending Tor from Network Adversaries: A Case Study of Network Path Prediction. In *Proceedings on Privacy Enhancing Technologies Symposium*, PoPETS '15, pages 171–187. De Gruyter, June 2015.

[26] Katharina Kohls, Kai Jansen, David Rupprecht, Thorsten Holz, and Christina Pöpper. On the Challenges of Geographical Avoidance for Tor. In *Network and Distributed System Security Symposium*, NDSS '19, San Diego, CA, USA, February 2019. The Internet Society.

[27] Katharina Kohls and Christina Pöpper. DigesTor: Comparing Passive Traffic Analysis Attacks on Tor. In *European Symposium on Research in Computer Security*, ESORICS '18, pages 512–530, Barcelona, Spain, September 2018. Springer.

[28] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services. In *USENIX Security Symposium*, USENIX '15, Washington, DC, USA, August 2015. USENIX Association.

[29] Hooman Mohajeri Moghaddam, Baiyu Li, Mohammad Derakhshani, and Ian Goldberg. SkypeMorph: Protocol Obfuscation for Tor Bridges. In *ACM Conference on Computer and Communications Security*, CCS '12, pages 97–108, Raleigh, NC, USA, October 2012. ACM.

[30] Steven J. Murdoch and George Danezis. Low-Cost Traffic Analysis of Tor. In *IEEE Symposium on Security and Privacy*, SP '05, pages 183–195, Oakland, CA, USA, May 2005. IEEE.

[31] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning. In *ACM Conference on Computer and Communications Security*, CCS '18, pages 1962–1976, Toronto, Canada, 2018. ACM.

[32] Milad Nasr, Amir Houmansadr, and Arya Mazumdar. Compressive Traffic Analysis: A New Paradigm for Scalable Traffic Analysis. In *ACM Conference on Computer and Communications Security*, CCS '17, pages 2053–2069, Dallas, TX, USA, October 2017. ACM.

[33] Rishab Nithyanand, Oleksii Starov, Adva Zair, Phillipa Gill, and Michael Schapira. Measuring and Mitigating AS-level Adversaries Against Tor. In *Network and Distributed System Security Symposium*, NDSS '16, San Diego, CA, USA, February 2016. The Internet Society.

[34] Rebekah Overdorf, Marc Juarez, Gunes Acar, Rachel Greenstadt, and Claudia Diaz. How Unique is Your .onion? An Analysis of the Fingerprintability of Tor Onion Services. In *ACM Conference on Computer and Communications Security*, CCS '17, pages 2021–2036, Dallas, TX, USA, October 2017. ACM.

[35] Andriy Panchenko, Fabian Lanze, and Thomas Engel. Improving Performance and Anonymity in the Tor Network. In *International Performance Computing and Communications Conference*, IPCCC '12, pages 1–10, Austin, TX, USA, 2012. IEEE.

[36] Fatemeh Rezaei and Amir Houmansadr. Tagit: Tagging Network Flows Using Blind Fingerprints. In *Proceedings on Privacy Enhancing Technologies Symposium*, PoPETS '17, pages 290–307. De Gruyter, 2017.

[37] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated Website Fingerprinting through Deep Learning. In *Network and Distributed System Security Symposium*, NDSS '18, San Diego, CA, USA, February 2018. The Internet Society.

[38] Anant Shah, Romain Fontugne, and Christos Papadopoulos. Towards Characterizing International Routing Detours. In *Asian Internet Engineering Conference*, AINTEC '16, pages 17–24, Bangkok, Thailand, November 2016. ACM.

[39] Fatemeh Shirazi, Milivoj Simeonovski, Muhammad Rizwan Asghar, Michael Backes, and Claudia Diaz. A Survey on Routing in Anonymous Communication Protocols. *ACM Computing Surveys (CSUR)*, 51(3):1–39, 2018.

[40] Robin Snader and Nikita Borisov. A Tune-up for Tor: Improving Security and Performance in the Tor Network. In *Network and Distributed System Security Symposium*, NDSS '08, San Diego, CA, USA, February 2008. The Internet Society.

[41] Qixiang Sun, Daniel R. Simon, Yi-Min Wang, Wilf Russell, Venkata N. Padmanabhan, and Lili Qiu. Statistical Identification of Encrypted Web Browsing Traffic. In *IEEE Symposium on Security and Privacy*, SP '02, pages 19–30, Berkeley, CA, USA, May 2002. IEEE.

[42] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. RAPTOR: Routing Attacks on Privacy in Tor. In *USENIX Security Symposium*, USENIX '16, pages 271–286, Washington, DC, USA, August 2016. USENIX Association.

[43] Henry Tan, Micah Sherr, and Wenchao Zhou. Data-Plane Defenses Against Routing Attacks on Tor. *Proceedings on Privacy Enhancing Technologies Symposium*, 2016(4):276–293, October 2016.

[44] The Tor Project. Tor Metrics, January 2019. https://metrics.torproject.org.

[45] Torspec. Tor Path Specification, February 2021. https://gitweb.torproject.org/torspec.git/tree/path-spec.txt.

[46] Chris Wacek, Henry Tan, Kevin S Bauer, and Micah Sherr. An Empirical Evaluation of Relay Selection in Tor. In *Network and Distributed System Security Symposium*, NDSS '13, San Diego, CA, USA, February 2013. The Internet Society.

[47] Gerry Wan, Aaron Johnson, Ryan Wails, Sameer Wagh, and Prateek Mittal. Guard Placement Attacks on Path Selection Algorithms for Tor. In *Proceedings on Privacy Enhancing Technologies Symposium*, PoPETS '19, pages 272–291. Sciendo, 2019.

[48] Tao Wang, Kevin Bauer, Clara Forero, and Ian Goldberg. Congestion-Aware Path Selection for Tor. In *Financial Cryptography*, FC '12, pages 98–113, Kralendijk, Bonaire, February 2012. Springer.

[49] Zachary Weinberg, Jeffrey Wang, Vinod Yegneswaran, Linda Briesemeister, Steven Cheung, Frank Wang, and Dan Boneh. StegoTorus: A Camouflage Proxy for the Tor Anonymity System. In *ACM Conference on Computer and Communications Security*, CCS '12, pages 109–120, Raleigh, NC, USA, October 2012. ACM.

Appendix

Figure 9 shows the detailed results of the exit prediction performance evaluation for the top 10 countries in terms of relay bandwidth (cf. Section 5.1.3).

Figure 10 shows bandwidth distributions of guards and exits for five countries (cf. Section 5.3.3).

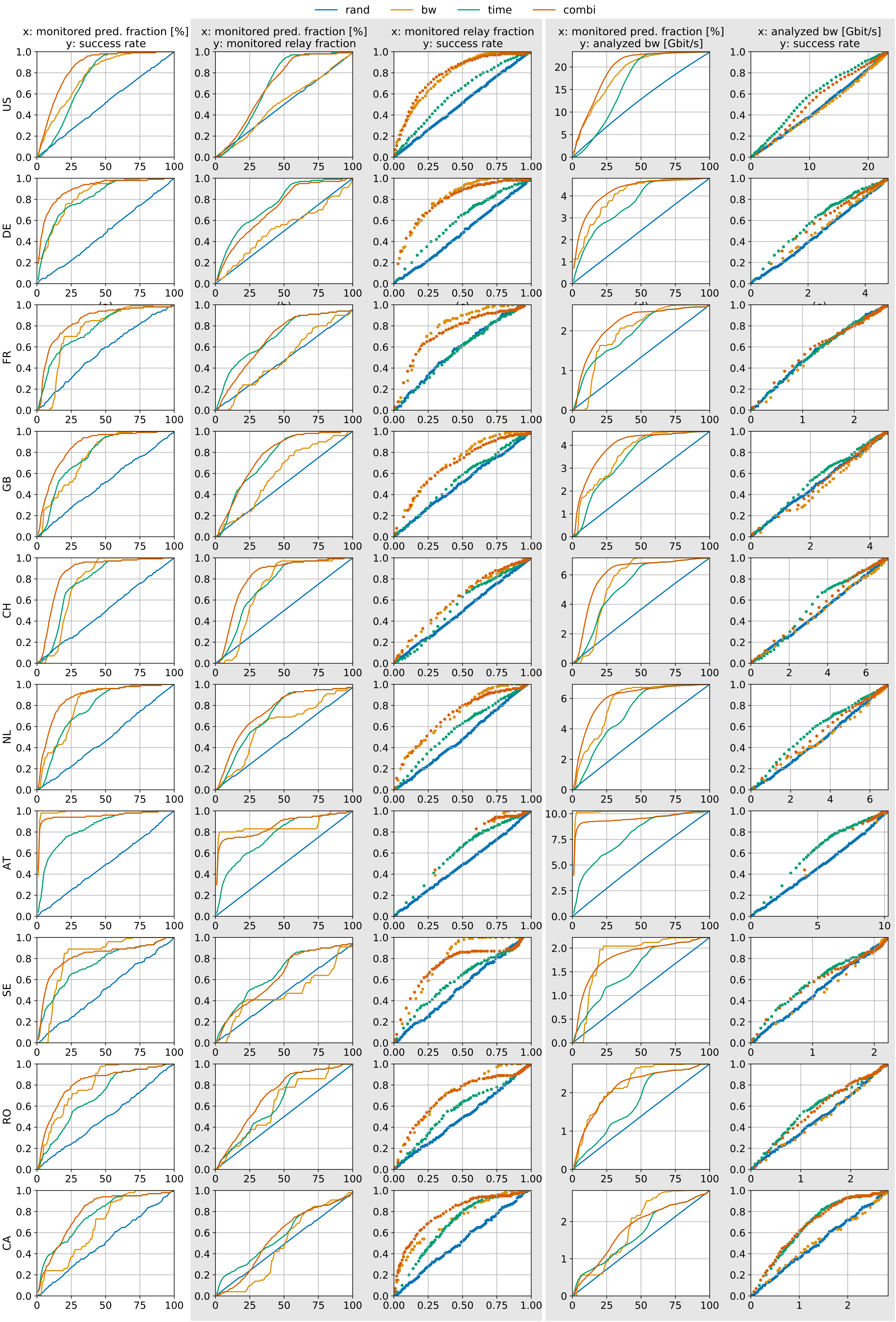

Figure 9. Detailed Exit Prediction Performance for the top 10 countries.

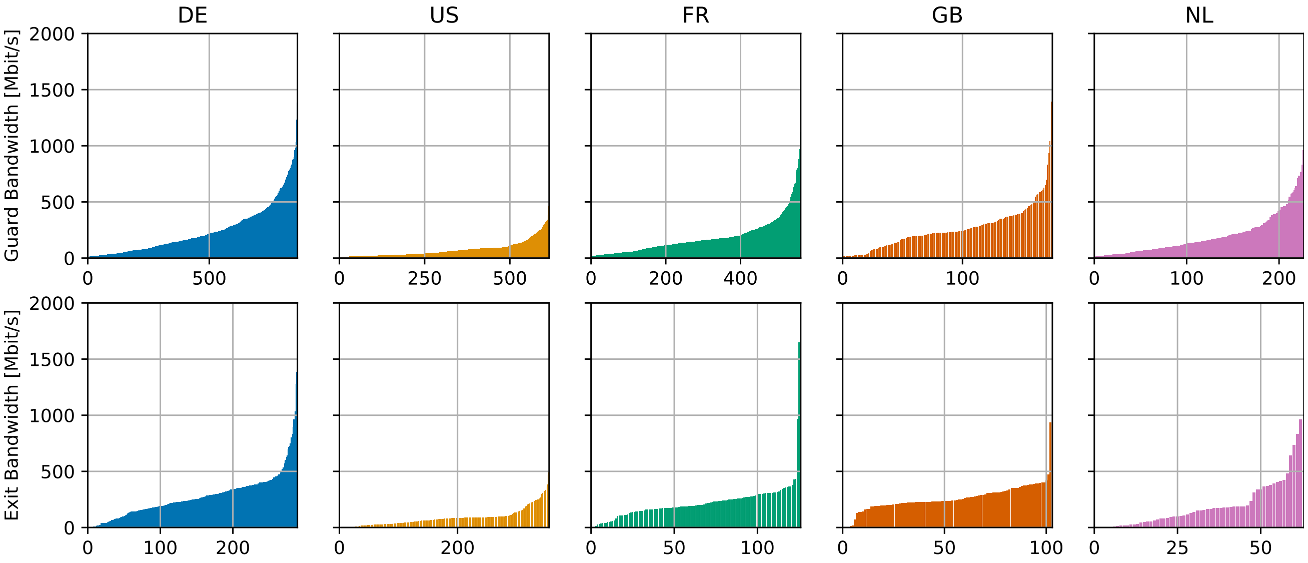

Figure 10. Individual relay bandwidths per country separated by Guards (top) and Exits (bottom).