sciendo

Proceedings on Privacy Enhancing Technologies ; 2019 (2):166–186

Hans Hanley*, Yixin Sun, Sameer Wagh, and Prateek Mittal

DPSelect: A Differential Privacy Based Guard Relay Selection Algorithm for Tor

Abstract: Recent work has shown that Tor is vulnerable to attacks that manipulate inter-domain routing to compromise user privacy. Proposed solutions such as Counter-RAPTOR [29] attempt to ameliorate this issue by favoring Tor entry relays that have high resilience to these attacks. However, because these defenses bias Tor path selection on the identity of the client, they invariably leak probabilistic information about client identities. In this work, we make the following contributions. First, we identify a novel means to quantify privacy leakage in guard selection algorithms using the metric of Max-Divergence. Max-Divergence ensures that probabilistic privacy loss is within strict bounds while also providing composability over time. Second, we utilize Max-Divergence and multiple notions of entropy to understand privacy loss in the worst-case for Counter-RAPTOR. Our worst-case analysis provides a fresh perspective to the field, as prior work such as Counter-RAPTOR only analyzed average case-privacy loss. Third, we propose modifications to Counter-RAPTOR that incorporate worst-case Max-Divergence in its design. Specifically, we utilize the exponential mechanism (a mechanism for differential privacy) to guarantee a worst-case bound on Max-Divergence/privacy loss. For the quality function used in the exponential mechanism, we show that a Monte-Carlo sampling-based method for stochastic optimization can be used to improve multi-dimensional trade-offs between security, privacy, and performance. Finally, we demonstrate that compared to Counter-RAPTOR, our approach achieves an 83% decrease in Max-Divergence after one guard selection and a 245% increase in worst-case Shannon entropy after 5 guard selections. Notably, experimental evaluations using the Shadow emulator shows that our approach provides these privacy benefits with minimal impact on system performance.

Keywords: Differential Privacy, Max-Divergence, BGP Hijack Attacks

DOI 10.2478/popets-2019-0025
Received 2018-08-31; revised 2018-12-15; accepted 2018-12-16.

***Corresponding Author: Hans Hanley:** Princeton University, E-mail: hhanley@princeton.edu

1 Introduction

Tor is an infrastructure for enabling anonymous communication. Tor is used by citizens, journalists, whistleblowers, businesses, and intelligence agencies throughout the world to protect privacy [9]. However, Tor's widespread use make it a high priority target. With over 7,000 relays transporting terabytes of traffic every day, Tor is a common target for attackers seeking to compromise clients' privacy anonymity[33].

Recently, researchers have found that Tor is particularly vulnerable to network-level adversaries [22, 30]. Network operators of Autonomous Systems (AS) can observe user traffic between clients and servers and use these observations to facilitate traffic analysis attacks against Tor users. While previous work on deanonymizing Tor clients has investigated threat models of passive network adversaries [15, 22], the recent work of Sun *et al.* [30] has shown that Tor clients are also highly vulnerable to active adversaries that manipulate inter-domain routing via Border Gateway Protocol (BGP) hijack attacks. An adversary AS can announce an equally-specific prefix to hijack client traffic causing a victim AS to be deceived into sending traffic to the adversary's AS. By announcing false Internet prefixes between ASes and by intercepting traffic from Tor clients, network operators can then deanonymize Tor clients. The 2014 Indosat attack showed the real-world possibility of these type of attacks; among the victims within the attacked AS were 44 Tor relays [17].

In Counter-RAPTOR [29], Sun *et al.* proposed a defense to equally-specific prefix BGP hijack attacks by directly favoring Tor guard relays with high resilience to these attacks. Certain Tor relays have high resilience to BGP attacks due to their position in the network relative to the client. Counter-RAPTOR's approach achieved an average resilience increase of 32%

Yixin Sun: Princeton University, E-mail: yixins@princeton.edu
Sameer Wagh: Princeton University, E-mail: swagh@princeton.edu
Prateek Mittal : Princeton University, E-mail: pmittal@princeton.edu

to BGP hijack attacks while maintaining similar performance to the currently deployed version of Tor with default settings (Vanilla Tor). The Counter-RAPTOR algorithm equally weights the relay's resilience to hijack attacks and the relay bandwidth. However, this results in a decrease in guard relay randomness, which in turn *leaks probabilistic information about client origin ASes.* A major drawback of their approach is thus a significant decrease in the Shannon entropy of client source ASes over time, allowing client ASes to be statistically fingerprinted [26, 36]. These fingerprinting attacks allow adversaries to link a client to her source AS. Adversaries capable of learning this information can more easily deanonymize clients (refer Section 3.1).

After 5 guard relay observations, the Shannon entropy for possible source ASes can decrease by as much as 61.6%. For adversaries interested in long-term deanonymization techniques, this opens another avenue to exploit Tor's vulnerabilities. Further, it can be relatively easy for an adversary to observe multiple guards in a shorter amount of time than ostensibly required (see Section 3). Understanding worst-case scenarios is thus imperative for practical deployments that utilize a client's location like Counter-RAPTOR. In this work, we show that using the Max-Divergence metric, we can gain a more holistic understanding of worst-case scenario behavior in Counter-RAPTOR and other proposals that utilize clients' locations. We further show that incorporating this differential privacy related metric into Counter-RAPTOR can ameliorate long-term deanonymization issues while also keeping Counter-RAPTOR's security properties.

Our contributions: In this work, we propose:

1. A way to quantify worst-case privacy loss in guard relay selection algorithms like Counter-RAPTOR. We utilize Max-Divergence (based on Differential Privacy) as our metric, which ensures that probabilistic privacy loss is within strict bounds while also providing composability over time.
2. A new guard relay selection algorithm, DPSelect, that outperforms Counter-RAPTOR by (a) reducing client vulnerability to fingerprinting attacks while (b) achieving similar resilience to BGP hijack attacks. We utilize the exponential mechanism (See Appendix A) to guarantee a worst-case bound on Max-Divergence/privacy loss. For the quality function used in the exponential mechanism, we show that a Monte-Carlo sampling-based method for stochastic optimization can be used to improve multi-dimensional trade-offs between security, privacy, and performance.
3. A quantification of information leaks (including temporal aspects) in Counter-RAPTOR's and DPSelect's guard selection algorithms using the metrics of Shannon Entropy, min-entropy, guessing entropy, and Max-Divergence from a worst-case perspective.
4. A comparative analysis of the security, privacy, and performance our new guard relay selection algorithm to Counter-RAPTOR and Vanilla Tor. Our approach achieves an 83% decrease in worst-case Max-Divergence and a 245% increase in worst-case Shannon entropy after 5 guard observations compared to Counter-RAPTOR for the month of October 2017.

Our approach to using ϵ-differential privacy for guard relay selection in Counter-RAPTOR can be applied to many different aspects of Tor. *Any Tor path selection algorithm dependent on the client's location can utilize the foundational aspects of our approach.* By selecting paths using differentially private mechanisms, the privacy loss of users can be rigorously quantified over time.

2 Background

We begin this section by giving a quick primer on Tor. We then describe Counter-RAPTOR's proposal for ameliorating Tor's weakness to BGP hijack attacks. Last, we review differential privacy and differentially private mechanisms.

2.1 Tor Overview

Tor is one of the most popular anonymous communication systems that protect users' privacy from other Internet users, service providers, and from network observers [9]. Around 7,000 volunteer-run relays called Onion Routers (OR) form the backbone of Tor and provide bandwidth and network connectivity for anonymous communications [33]. Tor clients select three relays viz., entry (guard), middle and exit relays to construct a circuit for accessing content.

Tor clients bias the selection of Tor relays by their individual bandwidths. Higher bandwidth relays have a higher probability of being selected while lower bandwidth relays have a lower probability. The Tor protocol also considers additional weights to these bandwidths in order to prevent congestion in the network [35].

Of particular importance is Tor client's guard relay since it is directly connected to the client. The guard selection protocol in the Tor network has evolved significantly from Tor's conception in 2004. Initially, because Tor clients pick relays according to their bandwidth for

each new circuit, a well-resourced adversary that participates in the Tor network could eventually deanonymize users by correlating traffic from the entry and exit nodes [4]. Tor first introduced guards in 2006 to counteract this threat. With this update, Tor clients only chose their guard/entry relay occasionally (order of months) rather than with each circuit [14]. Guard relay selection was revised again in 2015 to reduce information leakage caused by many guard selections in short periods of time due to changing network conditions [16, 34]. This update sought to balance resistance to active attacks that accelerated guard selection while trading off against other attacks. Simultaneously, it sought to maintain the performance and reliability of Tor network communication. As of November 2018, Tor clients select one guard from a set of three possible primary guards which have high availability [34]. This single guard is used for an average lifetime of 120 days and for a maximum of nine months [34]. Despite the changes made to guard selection, the threat of routing attacks, illustrated by RAPTOR [30], still persist.

2.2 Counter-RAPTOR

In this work, we build upon Counter-RAPTOR by Sun *et. al.* [29]. Counter-RAPTOR seeks to increase resilience to inter-domain routing attacks between Tor clients and their guard relays. Counter-RAPTOR increases overall resilience by selecting guards using the weighting function:

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

where $R(i)$ is a guard's resilience to BGP hijack attacks and B(i) is the guard's bandwidth. In Counter-RAPTOR, α determines how heavily bandwidth and resilience are weighted. In the recommended implementation of Counter-RAPTOR $\alpha = 0.5$.

Resilience Metric: If an adversary announces an equally-specific BGP prefix corresponding to the client's Tor guard, the client traffic could be routed to the adversary's AS. Consequently, the adversary could receive the traffic instead of the true origin AS [29]. The resilience $R(i)$ of a guard relay is the probability that a Tor client will not succumb to such a routing prefix attack if it is chosen as the guard relay. For more details on Counter-RAPTOR, refer to Sun *et al.* [29].

2.3 Differential Privacy

First proposed in 2006, differential privacy is a way of guaranteeing the privacy of individual members of a statistical database [11]. The main idea behind differential privacy is guaranteeing that any *single* database member's inclusion in a database does not change the output of any analysis in a *significant* way.

In differential privacy's model of computation, it is assumed that there exists some database D, comprised of n rows. A *query* is a function that is applied to the database to release information about the entries contained in the database D. A *privacy mechanism* $\boldsymbol{M}$ is an algorithm that takes the database D, a universe $\mathcal{X}$ of the possible data in D, and a set of queries as input and produces an output $\mathcal{O}$. If a privacy mechanism $\boldsymbol{M}$ can guarantee that the change in its output $\mathcal{O}$ due to the inclusion or removal of any single entry's data is bounded for any query, then it provides differential privacy.

Definition 1. ***(Differential Privacy):*** *A randomized mechanism $\boldsymbol{M}$ with domain $N^{|\mathcal{X}|}$ is ϵ-differentially private if for all $\mathcal{O} \in \text{Range}(\boldsymbol{M})$ and for all D_1, $D_2 \in N^{|\mathcal{X}|}$ such that D_1 and D_2 differ by at most one row, the following is true:*

$$\frac{Pr[\boldsymbol{M}(D_1) = \mathcal{O}]}{Pr[\boldsymbol{M}(D_2) = \mathcal{O}]} \leq e^{\epsilon} \tag{2}$$

The insight behind this formulation is that the ratio $Pr[\boldsymbol{M}(D_1) = \mathcal{O}]/Pr[\boldsymbol{M}(D_2) = \mathcal{O}]$ captures the privacy loss. Hence, a bound on this ratio can be used to quantify the privacy offered by a mechanism $\mathbf{M}$. If this bound holds for all possible outputs $\mathcal{O}$, then the mechanism is ϵ-differentially private.

2.3.1 Max-Divergence:

Another metric which is equivalent to differential privacy that is more instructive in this paper is Max-Divergence.

Definition 2. ***(Max-Divergence):*** *The Max-Divergence between two variables Y and Z taking values from the same domain $N^{|\mathcal{X}|}$ is defined as:*

$$D_{\infty}(Y \parallel Z) = \max_{S \subseteq \text{Range}(Y)} \left[\ln \frac{Pr[Y \in S]}{Pr[Z \in S]} \right] \tag{3}$$

Note that a mechanism $\boldsymbol{M}$ is ϵ-differentially private if and only if on every two neighboring databases x and y that differ in at most one element, $D_{\infty}(\boldsymbol{M}(x) \parallel \boldsymbol{M}(y)) \leq \epsilon$ and $D_{\infty}(\boldsymbol{M}(y) \parallel \boldsymbol{M}(x)) \leq \epsilon$. See Dwork [12] for a full proof. Differential privacy can therefore be defined using Equation 3. Throughout this work, we utilize the Max-Divergence to quantify the privacy offered by various algorithms.

3 Vulnerabilities in Counter-RAPTOR

In this section, we present our adversary model and analyze worst-case information leakage in the guard selection algorithm of Counter-RAPTOR. Even though Counter-RAPTOR considered the effect of entropy degradation in their guard relay selection algorithm, they only looked at the average case. We thus perform a thorough analysis of worst-case statistical attacks on Counter-RAPTOR using several well-established metrics and most notably Max-Divergence. Furthermore, we show that due to Tor network churn and active attacks the number of guard relay selections required for a statistical attack can occur in a shorter amount of time than ostensibly required.

For this analysis, we look at the top 93 most popular Tor client ASes [22]. These are the ASes where a large portion of Tor clients are located. We use Tor consensus data from October 21, 2017 [32] and October network topology data from the Center for Applied Internet Data Analysis (CAIDA) [5]. We also perform additional analysis using Tor consensus and CAIDA data from the entirety of 2017 to ascertain the sensitivity of our approach across time.

3.1 Adversary Model

In this work, we consider two different types of adversaries. We firstly consider a weak passive adversary that is interested in performing long-term deanonymization attacks. These deanonymization attacks make use of client guard selections that take place over a scale of months or years. The bulk of our analysis will focus on protecting against this passive adversary. We secondly consider an active adversary (BGP hijacker) capable of performing equally-specific IP prefix hijack attacks. This is the adversary that Counter-RAPTOR considered [29] and we also consider this adversary in our analysis of resilience to BGP hijack attacks (Section 5.1.4).

As noted, our analyses focus primarily on attacks on the anonymity of the client AS, the deanonymization of which we refer to as a fingerprinting attack. Fingerprinting attacks stem from location-based path selection approaches that leak probabilistic information about a client's source AS. Even though a single AS could serve thousands of Tor clients, identification of a Tor client's AS can be dangerous. As noted by Wails *et al.* [36], knowledge of a client's AS is problematic for three unique reasons: (1) The client AS can be targeted to divulge a user's real identity; (2) the diversity of a client's attributes (*e.g.* physical location) is much lower in a single AS and could be combined with auxiliary information to perform deanonymization; (3) the client AS can be used to link connections and profile Tor clients.

In this work, we focus on the most vulnerable users against whom fingerprinting attacks are the most successful. Given that users must trust the Tor network in order to use it regularly, even if a minority of Tor users can be reliably targeted and deanonymized, the attacks pose a risk to all Tor users. Thus, in this work, we highlight risks that the most vulnerable Tor users would have to endure.

3.2 Information Leakage in Counter-RAPTOR

Counter-RAPTOR and other location-based path selection algorithms leak information about clients' location. Here we examine information leakage in Counter-RAPTOR using Shannon entropy, min-entropy, guessing entropy, and most notably Max-Divergence.

Shannon Entropy: We first use Shannon entropy to evaluate information leakage in Counter-RAPTOR. Shannon entropy is a well-established metric for capturing the randomness of a system. Here, Shannon entropy considers the distribution of potential origin ASes for Tor clients after a guard selection. It corresponds to an uncertainty among 2^H possible choices where H is the Shannon entropy. As a result, a Shannon entropy of 6.54 bits implies that a client's origin AS is uniformly distributed among 93 source ASes [22]. Likewise, a Shannon entropy of 0 bits implies that a client's origin AS is uniquely identifiable. The Shannon entropy [6] of the client AS for a given guard relay selection is given by:

$$H(I) = -\sum_i p_i \log_2 p_i \tag{4}$$

where p_i is the probability that for a given relay, client i's AS is the initiator of the connection. The calculation of probability p_i is the same throughout Section 3.

In order to measure the decrease in Shannon entropy over multiple guard relay selections, we run 1000 simulations of multiple guard selections. In each simulation, we pick 50 different guards in succession using the probability distribution given by Counter-RAPTOR for AS5432.

Counter-RAPTOR's vulnerability increases over time as more guard relays are chosen. As seen in Fig. 1, as more guard relays are chosen, the average entropy decreases significantly from 6.54 bits. This is particularly

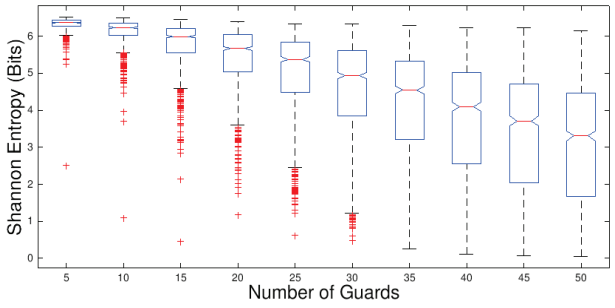

Fig. 1. Distribution of Shannon Entropy of Counter-RAPTOR clients in AS5432 over multiple guard observations in 1000 different simulations.

stark in the worst-case. After 5 guard relay selections, the worst-case Shannon entropy is near 2.51, a decrease of 61.6%. After 10 guard relay selections, the worst-case Shannon entropy is near 1-bit, an 81.8% decrease. This decrease means that an adversary can narrow the potential source to only 2 ASes (from the original 93).

Min-Entropy: We additionally look at min-entropy - a conservative metric to evaluate the privacy loss [26]. Like Shannon entropy, it also considers the distribution over possible source ASes and quantifies the effectiveness of a maximum likelihood estimator on the source AS of a Tor client. It is given by:

$$H_{Min}(I) = -\log_2 \max_i p_i \tag{5}$$

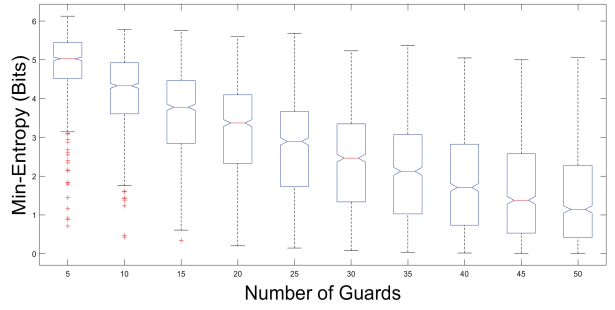

Fig. 2. Distribution of Min-Entropy of Counter-RAPTOR clients in AS5432 client AS over multiple guard observations in 1000 different simulations.

As seen in Fig. 2, the worst-case min-entropy nears 1-bit after 5 guard selection and goes near 0-bits after only 10 guard relay selections.

Guessing Entropy: We also evaluate the effectiveness of fingerprinting attacks using guessing entropy [26]. The guessing entropy corresponds to the average number of guesses with an optimum strategy required to determine the source AS of a client I. The guessing entropy is computed by rearranging the possible ASes in decreasing order of probability p_i and calculating:

$$H_{Guessing}(I) = \sum_i i \cdot p_i \tag{6}$$

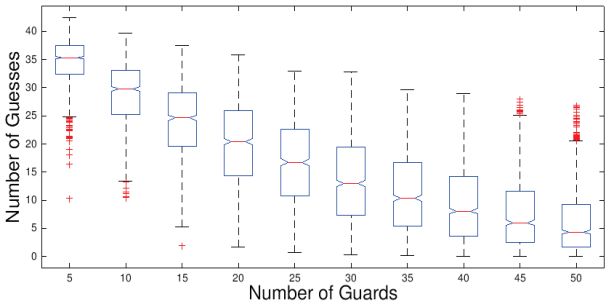

Fig. 3. Number of Guesses to determine client source AS of Counter-RAPTOR clients in AS5432 over multiple guard observations in 1000 different simulations.

As seen in Fig. 3, in the worst-case the guessing entropy decreases from 46 to under 10 after only 5 guard relay selections and to under 5 after 15 selections.

Max-Divergence/η-value: As stated before, we propose using the Max-Divergence of guard relays to understand the worst-case privacy loss from guard relay selection. Due to its composability property, the worst-case Max-Divergence for multiple selections is the sum of the Max-Divergence of each guard selection (see Definition 5). Hence, Max-Divergence is a more insightful metric to quantity the vulnerability of Counter-RAPTOR clients to statistical attacks over time. For purposes explained in Section 3, we refer to Max-Divergence as the η-value.

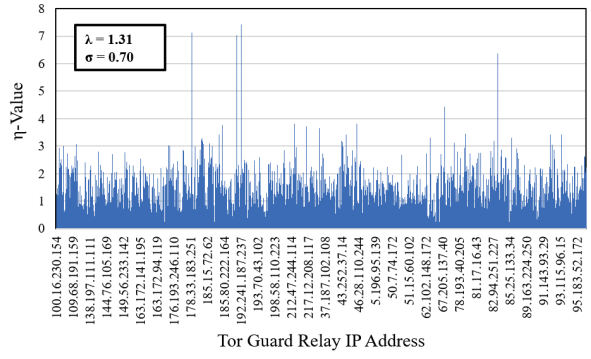

Fig. 4. Max-Divergence value across all possible Counter-RAPTOR Tor guard relays in the Top 93 Client ASes for the month of October 2017.

Using the Max-Divergence definition from Section 2.3.1, we can define the Max-Divergence/η-value

for any guard relay choice $\mathcal{G}$. The Max-Divergence is calculated by taking the *natural logarithm of the ratio of the highest probability of any client choosing a particular relay and the lowest probability of any client choosing a particular relay.*

In Fig. 4, even though the average Max-Divergence is 1.3, we see a divergent worst-case behavior. The largest Max-Divergence value exceeds 7 (for guard 192.241.187.237). This tells us that 192.241.187.237 in particular has a very skewed probability distribution amongst all the potential Tor client ASes.

Sensitivity of Information Leakage: Counter-RAPTOR makes use of the CAIDA network topology (changes monthly) and Tor consensus data (Changes hourly), which can invariably affect the information leakage over time. Thus, our analysis is dependent on changes in the Tor consensus data as well as the underlying Internet topology. As noted, these results depend on

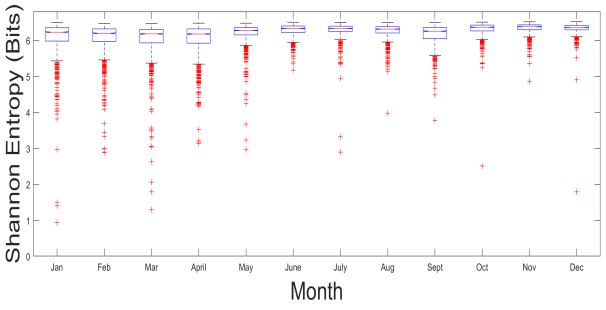

Fig. 5. Shannon-Entropy of Counter-RAPTOR clients in AS5432 client AS after 5 guard observations throughout 2017 in 1000 simulations.

the churn of Internet topology (this changes relatively slowly) and the change in AS distribution of guard relays (which also is low). In Fig. 5, we see that while there are changes in the distribution of leakage over a given month, the median entropy loss is similar. This shows that gradual changes to the network do not lead to significantly dissimilar results. However, Fig. 5 does indicate that over particular months, complex combinations of a particular network topology and Tor consensus data can lead to more unfavorable worst-case scenarios. This is further confirmed in the Max-Divergence analysis presented in Fig. 6. The Max-Divergence for each month is shown to be relatively consistent over time with only worst-behavior being variable. Max-Divergence elicits additional information about the vulnerability of users, which is not immediately available from entropy evaluations. For example, the worst case-scenario Max-Divergence/privacy-loss in the month of November at nearly 7 cannot be gleaned from the Shannon entropy analysis in Fig. 5.

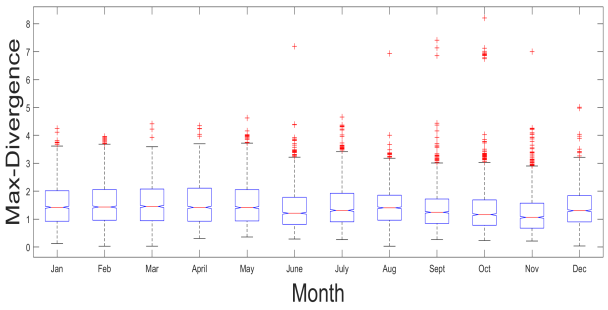

Fig. 6. Distribution of the Max-Divergence ratio across actives Tor guards in each month of 2017.

Summary of Vulnerability: As seen, various metrics show the vulnerability of Counter-RAPTOR to statistical client AS fingerprinting. The entropy metrics highlight the effectiveness of probabilistic statistical attacks. They show that even after 5 guard relay selections, Counter-RAPTOR, in the worst-case scenario, can divulge a great deal of client information. We also show that this vulnerability is persistent over the scope of a year and even peaks to worse values (than in our primary month of October 2017). The amount of information revealed by each selection can be seen most easily in the Max-Divergence analysis.

Although we focus on long-term attacks by global adversaries, because a new guard is only chosen periodically, these attacks appears challenging to mount at first glance. As of November 2018, from Tor specifications, guard lifetimes are 120 days [34]. Despite this, for a small percentage of users these attacks remain a glaring problem. Even though 5-10 guards can take a substantial amount of time to pick, long-term attacks remain feasible. We now discuss this issue in more detail.

3.3 Frequency of Guard Selection

Although Tor guard lifetimes are 120 days and Tor clients use all guards in their primary guard set before selecting a new one, the number of Tor guards used by a client over a given period of time can increase due to several passive and active factors [34]. For example, guards that are removed from the network or lose their guard flag can increase the number of guards selected. Active attacks include: (1) hibernation attacks that force guards to expend their bandwidth quota [22], and (2) DDoS attacks can force clients to select multiple guards over shorter time space. We briefly describe these issues below:

Guard Removal: The roster of Tor guards often changes as volunteers decide to remove their nodes from the network. As shown in Fig. 7, many guards are not available over long period of time. On average, only 60% of the guard relays are still available after 3.5 months. In the worst-case, clients must reselect a primary guard. Even for guards in the 99th percentile of bandwidth (which are known to be stable), almost 20% are offline within 2 months.

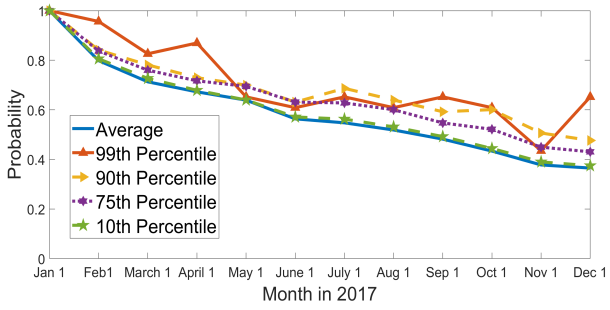

Fig. 7. Probability that guard a client picked on 01/01/17 is still in the network.

Active Attacks: Active attacks can also force Tor guard relays to go offline. Once offline these guards are labeled as unreachable (this status only lasts for a given amount of time before the guard is retried). This can cause clients to make use of another guard relay. To force a brand-new guard to be chosen an *attacker would need to perform an active attack on all guards that a client currently had in their guard list.*

Hibernation attacks cause client guards to go offline by downloading large files, expending their quota specified in its consensus. Network adversaries can also easily induce congestion or cause resource exhaustion. DDoS (Distributed Denial of Service) attacks targeted at specific Tor guards can increase the number of guards observed in a relatively short amount of time. As seen in Fig. 8, the amount of bandwidth required to simultaneously DDoS a series of Tor guards is within a reasonable range both for average and for the highest bandwidth Tor guards [32, 33].

Summary of Attack: We have shown that Counter-RAPTOR is vulnerable to fingerprinting after multiple guard selections. Further, we have shown that the amount of churn in the Tor network and active attacks reduce the time required for an adversary's observation of multiple client guards. At the same time, Tor clients often must re-select guards due to congestion, Out of Memory (OOM) errors [20], or guard failure. Ensuring worst-case bounds on information leakage in new guard relay selection algorithms (such as Counter-RAPTOR) is thus a high priority.

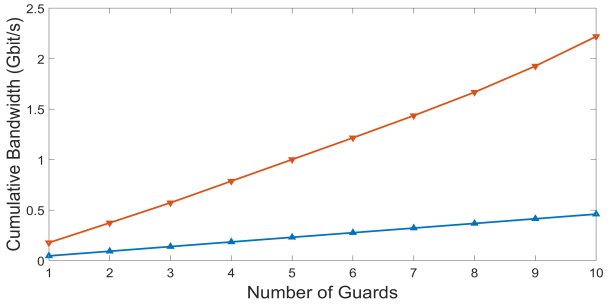

Fig. 8. Average and Top Cumulative Bandwidth of Tor guards after 10 guard selections

4 DPSelect: Differentially Private Tor Guard Relay Selection

We show in Section 3 that client location-dependent guard selection algorithms like Counter-RAPTOR are vulnerable to statistical fingerprinting attacks over time. In this section, we propose a new guard selection algorithm, DPSelect, that makes use of the Max-Divergence metric in combination with relay resilience and relay bandwidth to improve client privacy.

The following are the design goals of DPSelect:

1. *Protect against fingerprinting of Tor clients even in the worst-case over time.* DPSelect provides rigorous mathematical bounds on the privacy loss of guard selection choices and provides a guarantee on worst-case behavior against client fingerprinting attacks.
2. *Maintain Counter-RAPTOR's benefits of mitigating BGP hijack attacks on Tor.* DPSelect continues incorporating resilience into guard relay selection to protect clients from BGP hijack attacks.
3. *Provide comparable performance to Vanilla Tor.* DPSelect considers relay bandwidth to avoid causing excessive traffic congestion.

We now give an overview of the DPSelect algorithm and the details of our approach and its implementation.

4.1 DPSelect Algorithm

Tor relay selection algorithms generally assign a weight to each candidate guard relay and then probabilistically select a relay based on that weight. Our DPSelect algorithm incorporates the Max-Divergence metric based on ϵ-differential privacy into the weight function. We show that our algorithm provides a worst-case Max-Divergence/privacy-loss that can be set below any arbitrary threshold. Given that we use Max-Divergence

instead of conventional differential privacy to characterize the privacy, we use η instead of ϵ to avoid notational confusion.

Let Y be a random variable that determines which guard a given client in the network chooses, and Z be another random variable that determines which guard another client picks. The Max-Divergence of a given guard selection is defined as:

$$D_\infty(Y \parallel Z) = \max_{\mathcal{G} \in \text{Range}(Y)} \left[\ln \frac{Pr[Y = \mathcal{G}]}{Pr[Z = \mathcal{G}]} \right]$$

where $\mathcal{G}$ is any possible Tor guard relay. To find the above maximum, we need to determine which potential Tor client (which location) has the largest probability of choosing the given Tor guard $\mathcal{G}$ and which client (which location) has the least probability of choosing the given Tor guard $\mathcal{G}$.

The largest $D_\infty(Y \parallel Z)$ represents the worst-case η-value for any choice that a Tor client can make. To bound this value, we require a mechanism $\boldsymbol{M}$ that provides a worst-case max-divergence of η for every Tor guard relay choice across all possible Tor clients. We use the exponential mechanism (see Appendix A) to provide such a bound on the η-value/Max-Divergence.

Let x be the Tor client location in the network, i be the Tor guard relay being considered, and ϵ be a variable determining how private the selection should be. The exponential mechanism

$$\varepsilon_q^\epsilon(x, i) := e^{\epsilon q(x,i)} \tag{7}$$

where q is the quality function, provides a guarantee that the η-value is bounded by $2\epsilon\Delta q$. The sensitivity of the quality function Δq is defined as:
$\Delta q = \max_{r \in \mathcal{R}} \max_{\substack{x,y \in N^{|\mathcal{X}|} \\ \|x-y\|=1}} \mid q(x, r) - q(y, r) \mid$. See Appendix A for more details.

To choose a guard we define a weighting function, which is proportional to the probability that a given Tor guard is selected. The weight function using the exponential mechanism for each candidate guard relay i for a given client x is then defined as:

$$W(i) = e^{\epsilon q(x,i)} \tag{8}$$

4.2 DPSelect: Quality Function Optimization using Monte-Carlo Based Sampling

A naïve way to choose the quality function $q(x, i)$ in the exponential mechanism for a given client x and a candidate guard relay i is to use $q(x, i) = \alpha \cdot R(i) + (1-\alpha) \cdot B(i)$, where $R(i)$ is the resilience of relay i as defined in Counter-RAPTOR, and $B(i)$ is the normalized bandwidth of relay i. Using the exponential mechanism, the weight function then becomes:

$$W(i) = e^{\epsilon \cdot (\alpha \cdot R(i)^{x_1} + (1-\alpha) \cdot B(i)^{x_2})} \tag{9}$$

where $x_1 = 1$ and $x_2 = 1$. However, DPSelect should to seek to maintain high average bandwidths and resiliences. We thus seek to optimize this naïve quality function

First, we utilize a Monte-Carlo sampling-based method to stochastically optimize our results compared to the naive approach. The variables that are subject to change in the simulation are the α value, the resilience exponent (currently 1), and the bandwidth exponent (currently 1). In the Monte-Carlo simulation for the optimization function, we equally weigh the change in resilience and the change in bandwidth as shown below:

$$O(i) = 0.5 \cdot \Delta R(i) + 0.5 \cdot \Delta B(i) \tag{10}$$

We further need to specify the allowable privacy loss since this can be directly controlled with the exponential mechanism. We choose $\eta_{DPSelect} = 1.25$ to mirror the average η-value of Counter-RAPTOR ($\eta_{CR} = 1.3$).

4.2.1 Resilience Exponent and Bandwidth Exponent

We run a Monte-Carlo simulation for 2000 iterations (convergence) using the given optimization function (Equation 10).

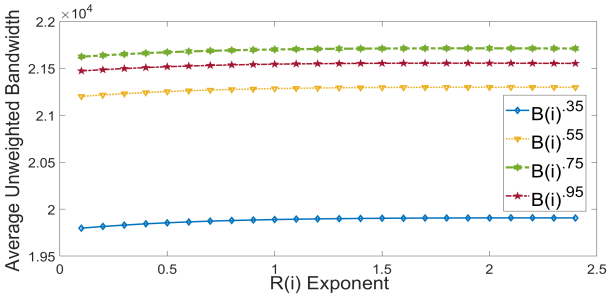

Fig. 9. Average unweighted bandwidth of Tor guard relays for top 93 Tor client ASes using consensus data from October 2017.

Here we show graphs illustrating the optimization that we performed and confirming its correctness. Fig. 9 shows the average unweighted bandwidth of guard relay selections with varying resilience exponents and bandwidth exponents. We can see that the average unweighted bandwidth reaches a maximum when the bandwidth exponent is 0.75, while the resilience exponent does not have a significant effect on the bandwidth. Thus, we use 0.75 as the optimal bandwidth exponent.

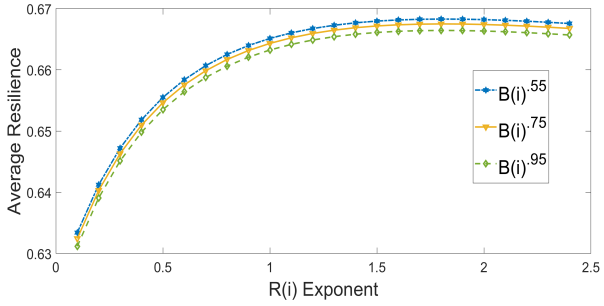

Fig. 10. Average resilience of top 93 Tor ASes to BGP hijack attacks vs the resilience exponent in weighting function for Tor guard relay using consensus data from October 2017.

Fig. 10 shows the average resilience of guard relay selections with varying resilience exponents and bandwidth exponents for top 93 Tor client ASes [22]. We can see that the average resilience reaches maximum when the resilience exponent is 2.0, while the bandwidth exponent does not have a significant effect. Thus, we use 2.0 as the optimal resilience exponent.

Overall, we find that the resilience exponent of 2.0 and the bandwidth exponent of 0.75 provide the optimal solution. The weight function then becomes:

$$W(i) = e^{\epsilon \cdot (\alpha \cdot R(i)^2 + (1-\alpha) \cdot B(i)^{0.75})} \tag{11}$$

In addition to these exponents, we saw that when $\alpha > 0.175$, there is a significant decrease in the average bandwidth of guard relay selections. As such, we choose α value to be 0.175 to ensure sufficient resilience to hijack attacks while not sacrificing too much bandwidth. The final recommended weighting function for DPSelect is then:

$$W(i) = e^{\epsilon \cdot (0.175 \cdot R(i)^2 + 0.825 \cdot B(i)^{0.75})} \tag{12}$$

We have published the Python code that we for stochastic optimization online for reproducibility [10].

4.2.2 ϵ-value based on worst-case η-value

As discussed in Appendix A on the exponential mechanism, the Max-Divergence of any selection is $2\Delta q\epsilon$. Here, our quality function $q(x, i) = \alpha \cdot R(i)^2 + (1-\alpha) \cdot B(i)^{0.75}$ for a given client x and a candidate relay i. Note that the bandwidth $B(i)$ of a relay does not change regardless of client locations and the Δq thus only depends on α and $R(i)$. Since $R(i)$ is in the interval $[0, 1]$, the Δq is bounded by α. Thus, the exponential mechanism provides $2\epsilon\alpha$-differential privacy.

With the recommended $\alpha = 0.175$ and our desired Max-Divergence bound $\eta_{DPSelect} = 1.25$, we have $\epsilon = 3.57$. This guarantees a worst-case η-value (Max-Divergence) of 1.25 for each guard relay selection. Note that $\eta_{DPSelect}$-value can be adjusted for the desired amount of privacy loss. For more details on the choice of the $\eta_{DPSelect}$-value, please see Appendix B.

4.3 DPSelect Implementation

DPSelect makes use of the same resilience metric used Counter-RAPTOR (see Sec. 2) which calculates resilience utilizing the Maxmind GeoIP database [24] and the CAIDA [5] to perform standard AS-path inference. Like Counter-RAPTOR, DPSelect assumes that clients are able to map IPs to ASN locally by using the Maxmind GeoIP database [24]. This database is already included in the Tor package and is used by Vanilla Tor. Our approach assumes that the CAIDA AS network topology for calculating AS paths for resilience is reliable and can be incorporated into the Tor browser [6]. The detailed steps of our Tor implementation are as follows (the initial steps are the same as Counter-RAPTOR [29]):

1. If the Maxmind ASN file and AS topology file have not been downloaded, the Tor client will download the two files from Maxmind and CAIDA, respectively, and save them in the local data directory. Otherwise, the Tor client will check if the local AS topology file is up to date (updated monthly), and if not, then download the latest version.
2. The Tor client will perform IP to ASN mapping, and compute the AS resilience R(i) of all candidate guard relays from the client AS the source AS.
3. The Tor client will compute a weight for each candidate relay using the current weighting function.
4. If the AS topology file was updated, the client will perform stochastic optimization to update the weighting function.
5. The Tor client will proceed with the guard selection. The remaining part of the circuit construction process stays the same as it is in Tor.

Because we use the same resilience metric as Counter-RAPTOR and do the stochastic optimization offline, DPSelect's computational overhead is nearly identical to Counter-RAPTOR. CAIDA updates the AS topology database monthly so the overhead of downloading the most recent file is low (< 700KB compressed). In our evaluation of 1000 randomly-selected client ASes, 90% finish the calculation in 0.6 seconds and all finish within 1 second.

5 Security and Performance Evaluations

In this section, we analyze the DPSelect algorithm and compare it with other guard selection algorithms. We

perform comprehensive (1) security and privacy evaluations and (2) performance evaluations. In the former, we quantify the privacy benefits of DPSelect while in the latter we analyze the performance overhead of DPSelect. In both, we comparatively analyze three different selection algorithms: Vanilla Tor, Counter-RAPTOR (with the recommended configuration $\alpha = 0.5$), and DPSelect.

5.1 Security and Privacy Evaluation

Here we evaluate the security and privacy of Tor guard relay selection algorithms from four perspectives:

1. **Entropy degradation:** Vulnerability of Tor clients to fingerprinting attacks using Shannon entropy, Min-entropy and Guessing entropy in the worst-case and over multiple guard selections.
2. **Max-Divergence:**Vulnerability of Tor clients to fingerprinting attacks using calculated η-values across all possible Tor guard relays.
3. **Client Anonymity:** Anonymity bounds for Tor clients using MATor [2].
4. **Resilience against BGP Hijack attacks:** Probability of Tor client ASes being resilient to a BGP hijack attack.

To calculate resilience and the probability distributions for each relay, we use Tor consensus data from October 21, 2017 [32] and the CAIDA network topology for October 2017 [5]. We also perform additional analysis using Tor consensus and CAIDA data from the entirety of 2017 to ascertain the sensitivity of our approach across time.

5.1.1 Vulnerability to Fingerprinting attacks

We first evaluate each algorithm's vulnerability to fingerprinting attacks. As noted, there is a trade-off between vulnerability to prefix hijack attacks and vulnerability to fingerprinting attacks. Note that Vanilla Tor is perfectly resilient to statistical fingerprinting based on guard selection (though it is vulnerable to hijack attacks). However, DPSelect algorithm, like many other client location-dependent relay selection algorithms such as Counter-RAPTOR, selects guard relays based on the client's location in the network. As a result, DPSelect is also vulnerable to client fingerprinting to some degree.

We presented and evaluated fingerprinting attacks using Shannon entropy, min-entropy, and guessing entropy on Counter-RAPTOR in Section 3. Here, we also evaluate the attack on DPSelect using these entropies and compare the results comparison with Counter-RAPTOR in terms of vulnerability. Similar to Section 3, we consider the top 93 Tor client ASes [22] as potential client locations.

Shannon Entropy: As noted in Section 3, Shannon entropy considers the distribution of potential source ASes of Tor connections (as computed by the attacker). The Shannon entropy [6] is calculated using Equation 4: $H(I) = -\sum_i p_i \log_2 p_i$ where p_i is the probability that for a given relay, client i's AS is the initiator of the connection. The calculation of probability p_i is the same throughout Section 5.

We run 1000 simulations where we pick 50 different guards in succession using the probability distribution created by DPSelect and Counter-RAPTOR for clients in AS5432 (one of the top 93 Tor ASes) [22]. We thus see the approximate distribution of Shannon entropy degradation for Tor clients in AS5432.

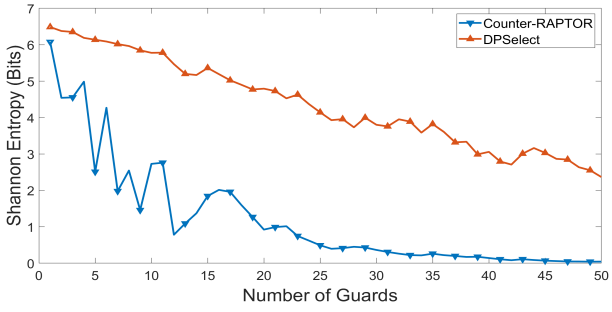

Fig. 11. Minimum Shannon Entropy of Counter-RAPTOR and DPSelect after multiple guard relay selections in 1000 simulations for clients in AS5432.

Fig. 11 shows the Shannon entropy degradation for both DPSelect and Counter-RAPTOR. We can see that in the worst-case, DPSelect provides better guarantees. After 6 guard choices, Counter-RAPTOR nears a worst-case Shannon entropy of 2 bits, while DPSelect maintains a worst-case Shannon entropy above 5 bits. DPSelect thus provides a better worst-case entropy for the attack proposed in Section 3. As noted in the Section 3, 5 guard choices can occur more quickly than ostensibly required. Due to guard removal, and active attacks, multiple guards can be selected over a shorter period of time.

Min-Entropy: To further evaluate DPSelect compared to Counter-RAPTOR, we consider min-entropy. As noted in Section 3, min-entropy is a more conservative metric for quantifying privacy loss than Shannon entropy [26]. To calculate min-entropy we used Equation 5: $H_{Min}(I) = -\log_2 \max p_i$.

Fig. 12 shows the result. We can see that DPSelect's worst-case min-entropy remains above 4 bits after 5 guard selections. This means that even after 5 relay

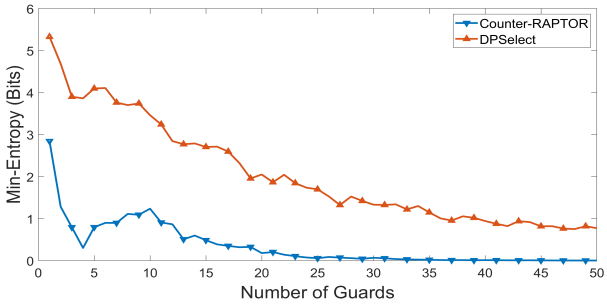

Fig. 12. Lowest Min-Entropy of Counter-RAPTOR and DPSelect after multiple guard relay selections in 1000 simulations for clients in AS5432.

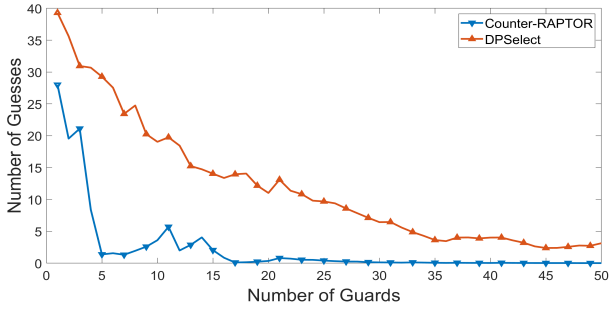

Fig. 13. Minimum Number of Guesses Entropy of Counter-RAPTOR and DPSelect after multiple guard relay selections over 1000 simulations for clients in AS5432.

selections, no adversary could trace the client source AS with greater than 6.25% confidence amongst the top 93 Tor ASes. In contrast, Counter-RAPTOR deteriorates rapidly. By 15 guard relays selections, the min-entropy is nearly 0. This means that an observing adversary could say with a high degree of confidence which AS initiated a particular connection.

Guessing Entropy: Guessing entropy quantifies the average number of guesses required for an adversary to identify the source AS of a client. The guessing entropy is calculated using the Equation 6: $H_{Guessing}(I) = \sum_i i \cdot p_i$. We can see in Fig. 13 that DPSelect's guessing entropy remains above 20 guesses after 150 guard selections. In contrast, Counter-RAPTOR's number of guesses goes below 5 guesses after 5 guard observations.

Summary: From the above evaluations, we can clearly see that DPSelect provides better worst-case guarantees against client fingerprinting attacks. Table 1 shows the complete results of Vanilla Tor, Counter-RAPTOR and DPSelect for all three entropies after 5 guard selections.

5.1.2 Max-Divergence (η-Value)

We assess the η-values (Max-Divergence) across all possible Tor guard relays for the three different guard

Algorithm	Shannon		Min		Guessing	
	H(·)	% Dec.	H(·)	% Dec.	H(·)	% Dec.
Vanilla Tor	6.54	0	6.54	0	6.54	0
Counter-RAPTOR	2.51	61.6	0.80	87.8	1.38	97.0
DPSelect	6.14	6.11	4.09	37.5	29.3	37.0

Table 1. Worst-case entropies for Vanilla Tor, Counter-RAPTOR, and DPSelect after 5 guard relay selections.

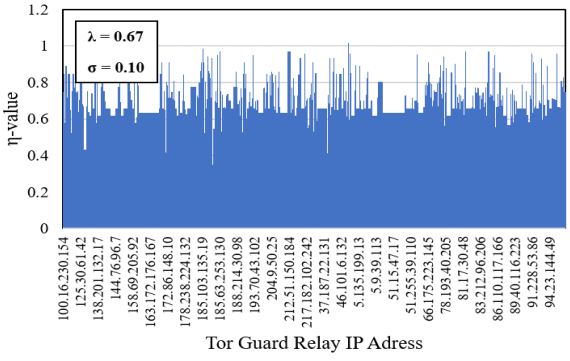

Fig. 14. η-values across all possible Tor guard relays for the month of October 2017 for DPSelect.

relay selection algorithms. Unlike the entropies, Max-Divergence can quantify the privacy loss caused by picking a particular relay across all 93 possible Tor ASes succinctly. The Max-Divergence is calculated by taking the natural log of the ratio of the highest probability of choosing a particular relay to the lowest probability of choosing a particular relay.

Vanilla Tor: In Vanilla Tor, each Tor client selects Tor guard relays with the same probability distribution. Vanilla Tor biases solely on the bandwidth of the given relay. Each client thus has the same probability of choosing each guard relay, regardless of the location of the client. Thus, for all possible guard relays $\mathcal{G}$ where X and Y are random variables specifying the guard relay choice for any two Tor clients, $\frac{Pr[X=\mathcal{G}]}{Pr[Y=\mathcal{G}]} = 1$. As a result, $\max \ln \frac{Pr[X=\mathcal{G}]}{Pr[Y=\mathcal{G}]} = 0$. The η-value for any choice is therefore 0 across all Tor ASes and for all Tor guard relay choices in Vanilla Tor.

Counter-RAPTOR: As discussed in Section 3, the worst-case η-value exceeds 7 for the month of October. Furthermore, because Counter-RAPTOR determines resilience values based on the network topology, even the maximum value of 7 is not an upper bound.

DPSelect: Similar to Counter-RAPTOR, each Tor client has a different probability of choosing each guard relay based on their AS location in DPSelect.

Fig. 14 shows the η-value across all possible Tor guard relays for DPSelect in October 2017. The average η-value is only 0.67, which is a 48% improvement

(decrease) from the average η-value (1.3) in Counter-RAPTOR. Furthermore, while 1.25 is the theoretical bound for worst-case η-value, the actual value is 1.05 in this case. Overall, DPSelect achieves an 83% decrease in the worst-case and a 48% decrease in the average-case η-value.

Note that in addition to the decrease in the η-value (Max-Divergence), DPSelect provides a worst-case guarantee on the η-value. The exponential mechanism ensures that the η-value (Max-Divergence) never exceeds 1.25. The properties of the exponential mechanism imply that this is true across all Tor client ASes and for all possible guard relay selections, in contrast to Counter-RAPTOR which provides no worst-case η-value.

η-Value Summary: Among the client location-dependent Tor relay selection algorithms, DPSelect provides vastly stronger guarantees. The worst-case η-value in the DPSelect approaches is dependent only on the chosen η-value and is independent of the Tor network topology. However, for Counter-RAPTOR, the worst-case η-value is dependent on the underlying network topology and thus can vary widely. Specific choices have a high amount of variance between even the top 93 ASes for Counter-RAPTOR resulting in a large Max-Divergence.

5.1.3 Sensitivity of Information Leakage

Like Counter-RAPTOR, DPSelect makes use of the CAIDA network topology and Tor consensus data. As a result, the entropy and Max-Divergence analysis performed above is sensitive to the time period in which it is conducted. We now consider the sensitivity of our analysis to changing network conditions by considering data over the course of 2017. Fig. 15 shows the distribution of Shannon-Entropy of DPSelect clients throughout 2017. We see that there are only minimal changes in the distribution of leakage, where worst-case entropy is consistently above 6 bits after 5 guard relay selections. Similarly, we can see consistent worst-case scenario behavior for Max-Divergence over time in Fig. 16 throughout 2017, where the worst-case Max-Divergence never exceeds 1.05. We thus see that our results are robust to changing network conditions.

5.1.4 Client Anonymity

We assess the anonymity of Tor clients of Vanilla Tor, Counter-RAPTOR, and DPSelect using MATor, a framework for evaluating the degree of anonymity in Tor with rigorously proved bounds [2]. MATor considers a given client and measures anonymity with reference to the different relays that may be chosen. MA-

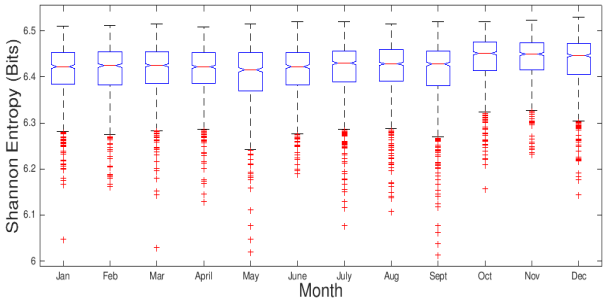

Fig. 15. Distribution of Shannon-Entropy of DPSelect clients in AS5432 client AS after 5 guard observations throughout 2017 in 1000 simulations.

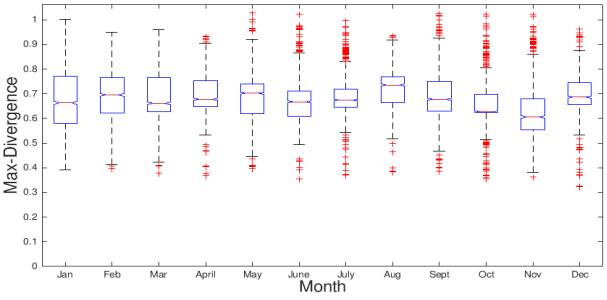

Fig. 16. Distribution of Max-Divergence for Tor guards throughout 2017.

Tor evaluates three anonymity notions: sender, recipient and relationship anonymity. For DPSelect, the recipient anonymity never changes because it does not alter exit relay selection. Thus, we do not show recipient anonymity in our evaluation.

We pick the top Tor client location AS6128 as the client AS and use MATor's default configuration with a multiplicative factor $\epsilon = 1.3$, ports setting of HTTPS+IRC vs. HTTPS, and 0.5% of total relay (33 of the 6600 available nodes) as compromised nodes (refer [2] for more details). We use Tor consensus files from 10/1/2017-10/5/2017 with server descriptors from October 2017 for our evaluation.

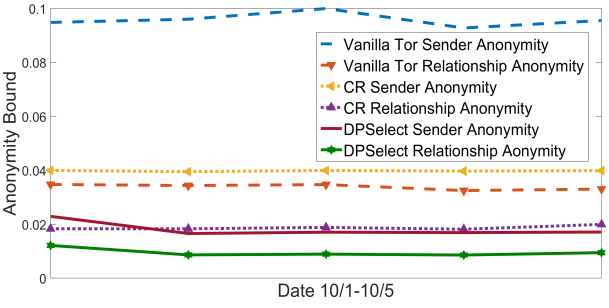

Fig. 17. MATor Anonymity Bound 10/1/2017 - 10/5/2017

The results in Fig. 17 show tighter bounds for the sender and relationship anonymity for both the DPS-

elect and Counter-RAPTOR selection algorithms compared to Vanilla Tor. This is due to both algorithms' additional dependence on relay resilience which skews the distribution away from placing trust in a smaller set of high-bandwidth nodes.

For DPSelect, we additionally see marginally better anonymity bounds compared to Counter-RAPTOR. This shows that DPSelect further redistributes and increases the set of relays in which trust is placed, providing better MATor anonymity guarantee.

5.1.5 Resilience against BGP Hijack attacks

The main goal of Counter-RAPTOR is to improve resilience of Tor clients to BGP hijack attacks. DPSelect seeks to provide similar improvements in resilience. Specifically, we evaluate and compare the average resilience for each of the top 93 Tor client ASes [22] for DPSelect and Counter-RAPTOR. Assuming $\mathcal{G}$ denotes the set of all guard relays, we calculate the resilience as:

$$\text{Resilience} = \sum_{i \in \mathcal{G}} P_{pick}(i) \cdot R(i) \tag{13}$$

Fig. 18 shows the probability of being resilient to hijack attacks for the top 93 Tor client ASes. We can see that Counter-RAPTOR and DPSelect provide similar improvements in resilience to BGP hijack attacks. While DPSelect does not provide as large an increase as Counter-RAPTOR, the difference is not significant. Our use of a Monte-Carlo simulation to optimize the resilience provided by DPSelect thus succeeded in ensuring that we maintained a high average resilience.

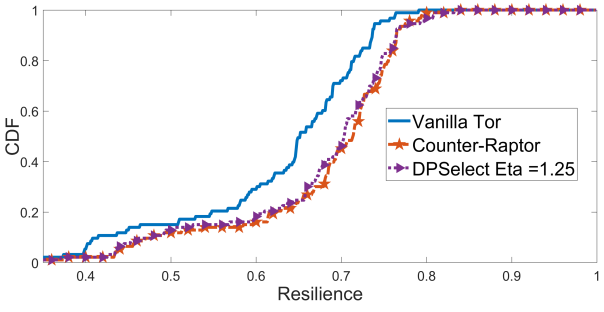

Fig. 18. CDF of the probability of being resilient to BGP hijack attacks for three different guard relay selection algorithms in October 2017 across the top 93 Tor client ASes.

5.2 Performance Evaluation

A major shortcoming of many guard relay selection algorithms that provide additional security or privacy is a decrease in performance. We evaluate and compare the performance of Vanilla Tor, Counter-RAPTOR and DPSelect from three aspects:

1. **Bandwidth:** The average bandwidth of a selected guard relay across the top 93 Tor client ASes.
2. **Load Balancing:** Load balancing amongst guards.
3. **Shadow Emulation:** Performance evaluation using Shadow Tor emulator [18].

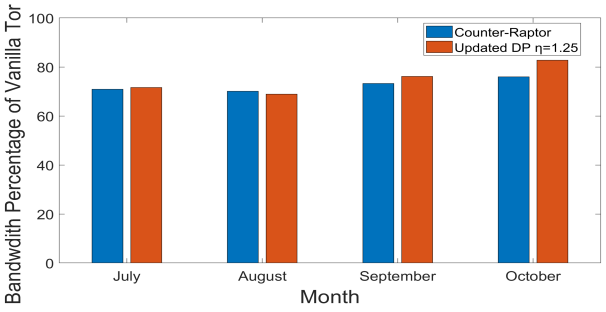

Fig. 19. Average bandwidth of guard relay selection choice for Counter-RAPTOR and DPSelect over the months of July, August, September, and October 2017.

5.2.1 Bandwidth

We first compute the average bandwidth of guard relay selections for the top 93 Tor client ASes [22]. Let $\mathcal{A}$ denote the Top 93 Tor ASes and $\mathcal{G}$ denote the set of all guard relays. We calculate the average bandwidth as:

$$\text{Average Bandwidth} = \frac{1}{93} \sum_{a \in \mathcal{A}} \sum_{i \in \mathcal{G}} P_{a,pick}(i) \cdot B(i) \tag{14}$$

Fig. 19 shows the average bandwidth as a percentage of Vanilla Tor's average bandwidth for Counter-RAPTOR and DPSelect from July to October 2017. Since both DPSelect and Counter-RAPTOR bias the guard selections with bandwidth *and* resilience, they both have lower average bandwidths compared to Vanilla Tor. However, we see that DPSelect maintains approximately the same average bandwidth over time as Counter-RAPTOR (at around 75% utilization compared to Vanilla Tor) while outperforming it each month (with the exception of August).

Our use of a Monte-Carlo simulation to optimize the bandwidth provided by DPSelect thus succeeded in ensuring that we maintained a high average bandwidth.

5.2.2 Load Balancing

DPSelect should ensure that it is not overloading any particular relay and is correctly load-balancing traffic amongst all possible Tor guard relays. Fig. 20 shows the bandwidth distribution of guard relay selections by Vanilla Tor, Counter-RAPTOR and DPSelect. DPSelect and Counter-RAPTOR have nearly the same load-balancing results. Both algorithms tend to bias slightly

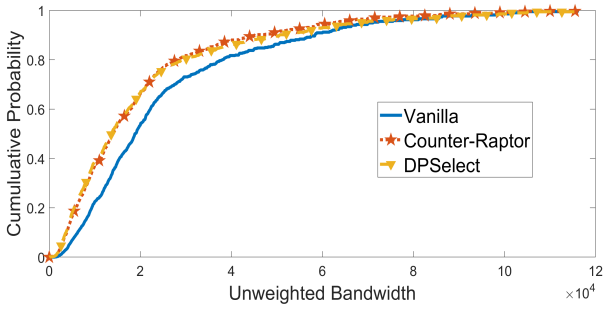

Fig. 20. Distribution of bandwidths of guard relays selected by Vanilla Tor, Counter-RAPTOR, and DPSelect for the October 21, 2017 Consensus.

more Tor traffic towards lower-bandwidth relays than Vanilla Tor due to the consideration of relay resilience.

5.2.3 Shadow Emulation

In order to validate DPSelect's ability to sustain a large throughput in an actual Tor network (large scale and whole system network performance), we analyze the performance of DPSelect using the Shadow emulator [18]. We implemented DPSelect in C based on Tor's original source code. We use the default network configuration in Shadow Tor as in Table 2.

Type	Number
Web Client	360
Bulk Client	40
Web Server	100
Guard Relay	14
Exit Relay	10
Guard/Exit Relay	5
Middle Relay	66

Table 2. Tor network configuration in Shadow emulator

Because both Counter-RAPTOR and DPSelect algorithm are client location-dependent, we need to assign meaningful IP addresses to all the client nodes. We use the IP addresses in the default Shadow configuration file for Tor relays and we uniformly chose IP addresses from the top 93 Tor client ASes [22] for the 400 Tor clients in the simulation.

Figs. 21– 24 show the network performance of Vanilla Tor, Counter-RAPTOR, and DPSelect. As can be seen, the performance for all three is comparable throughout the simulation. Figs. 21 and 22 in particular show the 60 second average receiver and sender throughput for all Tor nodes. Counter-RAPTOR and DPSelect have similar performance. However, in both graphs from ticks 2500-3000, DPSelect slightly outperforms Counter-RAPTOR.

As seen in the download times for the 320KB data in Fig. 23, Vanilla, Counter-RAPTOR, and DPSelect have nearly identical performances. For the 5MB download times in Fig. 24, DPSelect has slightly worse performance than Counter-RAPTOR and Vanilla Tor. However, the difference is not significant.

5.3 Summary of Results:

The DPSelect algorithm:

1. Improves significantly the worst-case Shannon entropy, min-entropy, and guessing entropy after 5 guard relay selections, thus defending against the fingerprinting attack outlined in Section 3.
2. Provides a worst-case guarantee on the η-value using the definition of Max-Divergence (whereas Counter-RAPTOR has no guarantee).
3. Provides comparable but slightly less resilience to BGP hijack attacks compared to Counter-RAPTOR.
4. Achieves comparable bandwidth and load-balancing results to Counter-RAPTOR and similar Shadow simulated performance to Counter-RAPTOR and Vanilla Tor.

6 Related Work

Here we discuss past works on defenses to network-level adversaries on the Tor network and recent attacks on these various proposed defenses.

Defenses against Network-level Adversaries. Network-level adversaries on the Tor network are a studied topic [13, 15, 22, 27, 30]. Previous works have proposed various Tor relay selection algorithms to defend against such network-level adversaries.

Akhoondi *et al.* proposed LASTor [1], which takes into account AS-level path and relay locations in selecting Tor relays to avoid having the same AS appear on both sides of the circuit. Johnson *et al.* proposed TAPS [21] which uses clustering around representative ASes that are hubs of Tor traffic before guard selection. Barton *et al.* [3] proposed DeNASA, which avoids a list of suspect ASes when constructing circuits in advance. Lastly, Nithyanand *et al.* proposed Astoria [28], which considers relay capacity, asymmetric routing, and colluding ASes in path selection. Astoria leverages recent developments in network measurement to perform path prediction and intelligent relay selection, while load balancing across the Tor network to prevent relay overload. Unlike our approach which seeks to protect against active BGP attacks while also preventing information leakage, Astoria focuses primarily on protecting against

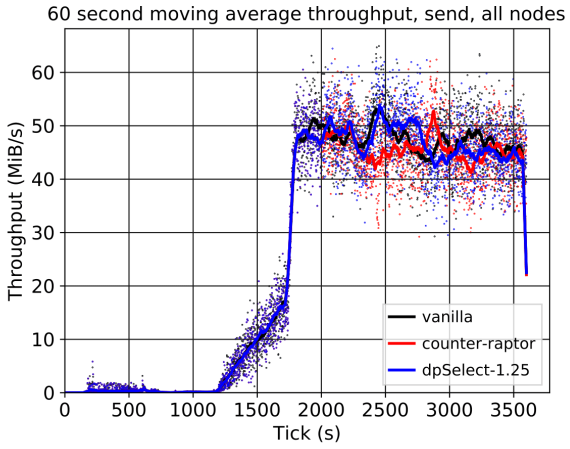

Fig. 21. 60 second average receiving throughput for all nodes.

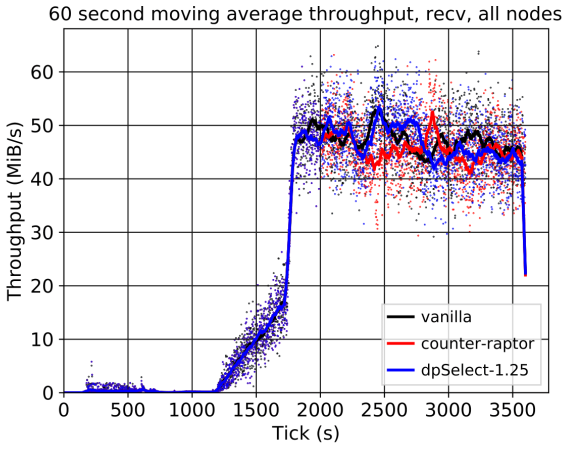

Fig. 22. 60 second average throughput for all nodes.

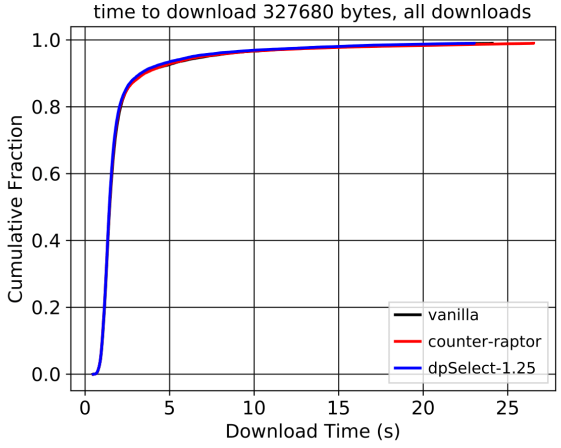

Fig. 23. Download time for 320KB data.

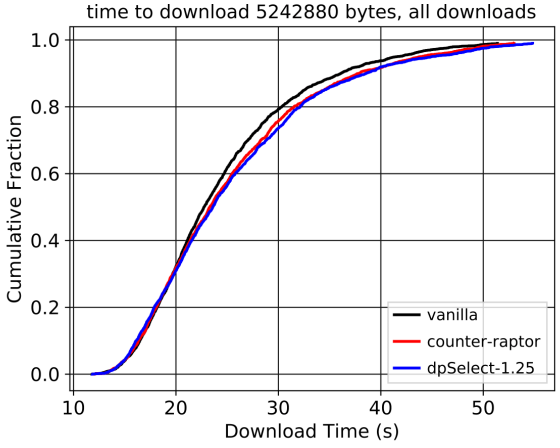

Fig. 24. Download time for 5MB data.

correlation attacks by AS-level adversaries [28]. However, unlike Vanilla Tor, (but like DPSelect) all these proposed relay selection algorithms are client location-dependent, which expose them to fingerprinting attacks from adversaries who can observe the clients' guard relay selections over time.

In contrast to the above approaches, Tan *et al.* [31] make use of data-plane defenses to detect routing interception attacks. Their approach utilizes several statistical invariants within the Tor network as well as periodic AS-neighbor discovery to detect routing attacks. By alerting clients of which guards are not under attack using a SafeGuard flag, Tan *et al.* [31] seeks to ensure that clients do not use compromised circuits. This is in contrast to Counter-RAPTOR [29] and DPSelect which seek to disable adversaries from being able to perform BGP hijack attacks at all.

Temporal Attacks on Tor. Recent research has also begun to look at anonymity degradation over time. Wails *et al.* [36] proposed Tempest attacks and showed how client mobility, usage patterns, and changes in the network topology over time affect the privacy of Tor users. In particular, Tempest makes use of mean Shannon entropy degradation over time to evaluate the vulnerability of Counter-RAPTOR to client fingerprinting attacks. This is in contrast to our approach which focuses on the worst-case scenario by utilizing a comprehensive set of entropy metrics. Other attacks on the Tor network also examine user behavior over time. Intersection attacks like those proposed by Danezis and Serjantov [7] make use of statistical disclosures when users send messages into mix networks. Danzeis and Tronsosco [8] have also shown how to use Bayesian inference to deanonymize persistent communications.

Differential Privacy in Tor. Jansen *et al.* [19] proposed to use differential privacy in order to safely measure Tor user behavior and to calibrate user parameters. They used differential privacy to prevent deanonymization of specific users given the output of their data collection. However, our work is the first to incorporate differential privacy into Tor relay selection to protect users from fingerprinting attacks.

7 Future Work and Limitations

We outline several directions for future work and the limitations of our approach here.

7.1 Future Work

Other differentially-private mechanisms. In our work, we adopt the exponential mechanism to provide the worst-case guarantee for DPSelect. However, there are other mechanisms which are yet to be explored. Use of the Laplace Mechanism and the Gauss Mechanism to add noise to the probability distribution of guard choices is an obvious future step. R-fold approximate differential privacy (ADP) also offers an alternative well-established framework and mechanism for characterizing privacy loss after r observations [25]. It will be interesting to compare different mechanisms in terms trade-offs between resilience to hijack attacks, client privacy, and performance.

Other differentially-private composition methods. In our work, we only consider linear worst-case composition. Advanced composition methods could be used to obtain a more graceful privacy decay [12]. By assuming a normal noise distribution, the moment accountant method discussed by Abadi *et al.* [23] could be used to track privacy loss. Meiser *et al.* have also shown that the bounds of privacy loss in ADP can be improved by making use of a numerical method called privacy buckets to capture privacy loss [25].

Evaluation on other client location-dependent relay selection algorithms. We primarily focus on demonstrating our entropy-based metrics for Counter-RAPTOR in Section 3. These metrics are generalizable and can be used to evaluate the information leakage for other client location-dependent relay selection algorithms as well, such as LasTor [1], Astoria [28], TAPS [21] and DeNASA [3]. We leave these evaluations for future work.

7.2 Limitations

Our use of Counter-RAPTOR's resilience metric in our quality function leads to several limitations that we now address.

As noted in Section 2, the resilience metric is averaged over each potential adversary AS. An *equally-specific* BGP prefix attack from a *well-chosen* location could be more effective than the resilience indicates. We leave the development of more refined metrics as an interesting direction of future work.

Counter-RAPTOR's resilience metric only takes into account the probability of being resistant to *equally-specific* BGP hijack attacks. Sub-prefix hijack attacks are ostensibly still an issue. However, *sub-prefix* hijack attacks generally affect the whole Internet, and thus they are less stealthy and can be more easily detected. One approach to proactively mitigate sub-prefix attacks is to move Tor relays into /24 prefix blocks [29].

8 Conclusion

Since the discovery of Tor's vulnerability to BGP hijack attacks, several defenses that choose guard relays non-uniformly have been proposed. Counter-RAPTOR [29] fixed this issue by weighting guard relay selection based on resilience to BGP hijack attacks as well as bandwidth. However, while Counter-RAPTOR is less vulnerable to BGP hijack attacks, it is more vulnerable to statistical fingerprinting attacks. We motivate our work by showing that adversaries can utilize passive attacks to identify client ASes. We then demonstrate that in the worst-case, after 5 guard relay selections, the Shannon entropy of the potential origin AS among the top 93 Tor ASes [22] can reduce to as low as 2.5 bits.

Our work demonstrates the benefits of using a differentially private approach to guard relay selection in Tor. Our approach shows that choosing a guard relay based upon client location, while providing rigorous bounds on information leakage and reasoning about privacy loss over time is feasible. By utilizing the DPSelect guard selection algorithm for $\eta = 1.25$, we record better security and privacy compared to Counter-RAPTOR. Specifically, we achieve an 83% decrease in worst-case Max-Divergence, and 245% improvement in worst-case Shannon entropy after 5 guard relay selections. Simultaneously, we achieve similar increases in resilience to BGP hijack attacks as Counter-RAPTOR.

To summarize, we show that differential privacy can be used to provide robust security and privacy guarantees to non-uniform guard relay selection algorithms while preserving performance and allowing added security such as resilience to BGP hijack attacks. We hope that our work can open up new lines of research that will allow a better understanding of the privacy of future guard relay selection algorithms.

9 Acknowledgements

We would like to thank the anonymous reviewers for helpful feedback on earlier versions of this paper. This work was supported by the National Science Foundation under the grants CNS-1704105, CNS-1553437, CNS-1617286, and the Army Research Office Young Investigator Program (ARO YIP) award.

References

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

[2] Michael Backes, Aniket Kate, Sebastian Meiser, and Esfandiar Mohammadi. (nothing else) MATor (s): Monitoring the anonymity of Tor's path selection. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 513–524. ACM, 2014.

[3] Armon Barton and Matthew Wright. DeNASA: Destination-Naive AS-Awareness in Anonymous Communications. *Proceedings on Privacy Enhancing Technologies*, 2016(4):356–372, 2016.

[4] Alexand Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. Trawling for tor hidden services: Detection, measurement, deanonymization. In *Security and Privacy, 2013 IEEE Symposium on*, pages 80–94. IEEE, 2013.

[5] CAIDA Internet topology map. https://www.caida.org/research/topology/.

[6] Ronald R Coifman and M Victor Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on information theory*, 38(2):713–718, 1992.

[7] George Danezis and Andrei Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In *International Workshop on Information Hiding*, pages 293–308. Springer, 2004.

[8] George Danezis and Carmela Troncoso. You cannot hide for long: De-anonymization of real-world dynamic behaviour. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, pages 49–60. ACM, 2013.

[9] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. *Proceedings of the 13th USENIX Security Symposium*, 2004.

[10] DPSelect Code. https://github.com/DPSelectro/DPSelect.

[11] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[12] Cynthia Dwork, Aaron Roth, et al. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

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

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

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

[16] Jamie Hayes and George Danezis. Guard Sets for Onion Routing. *Proceedings on Privacy Enhancing Technologies*, 2015(2):1–16, 2015.

[17] Hijack event today by Indosat. http://www.bgpmon.net/hijack-event-today-by-indosat/.

[18] Rob Jansen and Nicholas Hooper. Shadow: Running Tor in a box for accurate and efficient experimentation. *Network and Distributed System Security Symposium*, 2012.

[19] Rob Jansen and Aaron Johnson. Safely measuring Tor. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1553–1567. ACM, 2016.

[20] Rob Jasen, Florian Tschorsch, Aaaron Johnson, and Bjorn Scheuermann. The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network. In *The Network and Distributed System Security Symposium*, 2014.

[21] 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. *Network and Distributed System Security Symposium*, 2017.

[22] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users get routed: Traffic correlation on Tor by realistic adversaries. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & Communications security*, pages 337–348. ACM, 2013.

[23] Abadi Martin, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[24] Maxmind GeoLite ASN database. http://dev.maxmind.com/geoip/legacy/geolite/.

[25] Sebastian Meiser and Esfandiar Mohammadi. Tight on budget? tight bounds for r-fold approximate differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 247–264. ACM, 2018.

[26] Prateek Mittal and Nikita Borisov. Information leaks in structured peer-to-peer anonymous communication systems. *ACM Transactions on Information and System Security (TISSEC)*, 15(1):5, 2012.

[27] Steven J Murdoch and Piotr Zieliński. Sampled traffic analysis by internet-exchange-level adversaries. In *International Workshop on Privacy Enhancing Technologies*, pages 167–183. Springer, 2007.

[28] Rishab Nithyanand, Oleksii Starov, Adva Zair, Phillipa Gill, and Michael Schapira. Measuring and mitigating AS-level adversaries against Tor. *Network and Distributed System Security Symposium*, 2016.

[29] Yixin Sun, Anne Edmundson, Nick Feamster, Mung Chiang, and Prateek Mittal. Counter-RAPTOR: Safeguarding Tor Against Active Routing Attacks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 977–992. IEEE, 2017.

[30] 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*, pages 271–286, 2015.

[31] Henry Tan, Micah Sherr, and Wenchao Zhou. Data-plane Defenses against Routing Attacks on Tor. In *9th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2016)*, 2016.

[32] Tor consensus. https://collector.torproject.org/recent/relay-descriptors/consensuses/.

[33] Tor metrics. https://metrics.torproject.org/.

[34] Tor Guard Specification. https://gitweb.torproject.org/torspec.git/tree/guard-spec.txt.
[35] Tor Protocol Specification. https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt.
[36] Ryan Wails, Yixin Sun, Aaron Johnson, Mung Chiang, and Prateek Mittal. Tempest: Temporal Dynamics in Anonymity Systems. *Proceedings on Privacy Enhancing Technologies ; 2018 (3):22–42*, 2018.

A Differential Privacy

Here we present several differential privacy concepts that are used extensively throughout this work.

A.1 The Exponential Mechanism:

Particular randomized mechanisms $\boldsymbol{M}$ can provide ϵ-differential privacy. The exponential mechanism, specifically, is an ϵ-differentially private way of selecting one element from a set. Instead of adding noise directly to a database, it provides a way of returning randomized query results to maximize utility and to minimize the privacy loss. The exponential mechanism was designed for situations where the "best" response is desired.

Definition 3. ***(Exponential Mechanism):*** *Given some range $\mathcal{R}$, the exponential mechanism is defined with respect to a quality function $q : N^{|\mathcal{X}|} \times \mathcal{R}$, which maps database/output pairs to quality scores. q(x, r) measures how desirable an outcome r would be on the database x. Given a quality score $q : N^{|\mathcal{X}|} \times \mathcal{R} \to \mathbb{R}$, the exponential mechanism is defined as:*

$$\varepsilon_q^{\epsilon}(x, r) := e^{\epsilon q(x,r)} \tag{15}$$

The exponential mechanism is thus exponentially more likely to pick higher quality outcomes. The probability that r is selected thus increases exponentially with the value of $q(x, r)$.

Definition 4. ***(Sensitivity):*** *The sensitivity Δq of the quality function determines the privacy loss guaranteed by the exponential mechanism. The sensitivity of the quality function $q : N^{|\mathcal{X}|} \times \mathcal{R} \to \mathbb{R}$ is defined as:*

$$\Delta q = \max_{r \in \mathcal{R}} \max_{\substack{x,y \in N^{|\mathcal{X}|} \\ \|x-y\|=1}} \mid q(x, r) - q(y, r) \mid \tag{16}$$

Note that this sensitivity Δq is measured only with respect to the different database entered. Because the probability that $r \in \mathcal{R}$ is chosen increases exponentially $e^{q(x,r)}$, the intuition is that the *privacy loss* is bounded in the following way:

$$\ln\left(\frac{e^{\epsilon q(x,r)}}{e^{\epsilon q(y,r)}}\right) = \epsilon[q(x, r) - q(y, r)] \leq \epsilon \Delta q \tag{17}$$

By this definition (taking into account an addition normalization term), the exponential mechanism ensures $2\epsilon\Delta q$-differential privacy where Δq is the range over which $q(x, r)$ varies. For a full proof refer to Dwork [12].

A.1.1 Composition

Definition 5. ***(Composition):*** *Let $\mathcal{M}_1 : N^{|\mathcal{X}|} \to \mathcal{R}_1$ be an ϵ_1-differentially private algorithm and let $\mathcal{M}_2 : N^{|\mathcal{X}|} \to \mathcal{R}_2$ be an ϵ_2-differentially private algorithm. Then their combination defined as $\mathcal{M}_{1,2} : N^{|\mathcal{X}|} \to \mathcal{R}_1 \times \mathcal{R}_2$ by the mapping: $\mathcal{M}_{1,2}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x))$ is $\epsilon_1 + \epsilon_2$-differentially private.*

See Dwork [12] for more details.

B Varying η-Value in DPSelect Algorithm

The choice of the η-value (Max-Divergence) used in DPSelect is a societal choice. In Section 5, we chose $\eta = 1.25$ to compare closely with Counter-RAPTOR where the average η-value for was 1.3. Varying the η-value leads to multi-dimensional trade-offs in terms of privacy, security, and performance and thus needs to be chosen carefully. We conduct both the (1) security and privacy evaluation and (2) performance evaluation for different η-values.

B.1 Varying η-Value: Security and Privacy Evaluation

Here we evaluate the security and privacy of DPSelect for different η-values. We look at security and privacy from three different perspectives. Note that we do not discuss Max-Divergence in this section because we only change the η-value which directly corresponds to Max-Divergence, thereby rendering a discussion on Max-Divergence redundant.

1. **Vulnerability to Fingerprinting attacks:** Vulnerability of Tor clients to fingerprinting attacks using Shannon entropy, min-entropy and guessing entropy in the worst-case and over multiple guard selections.
2. **Client Anonymity:** Anonymity bounds for Tor clients using MATor [2].
3. **Resilience against BGP Hijack attacks:** Probability of Tor Client ASes being resilient to a BGP attack.

B.1.1 Vulnerability to Fingerprinting attacks

As in Section 5, to measure the decrease in the different entropy over multiple guard relay selections, we

run 1000 simulations for each DPSelect configuration in which we pick 50 different guards in succession. As in Section 5, we use the probability distribution for AS5432.

Shannon Entropy: The Shannon entropy [6] of a client I is calculated using Equation 4: $H(I) = -\sum_i p_i \log_2 p_i$ where p_i is the probability that for a given relay and client I is the initiator of the connection. The calculation of probability p_i is the same throughout Appendix B.

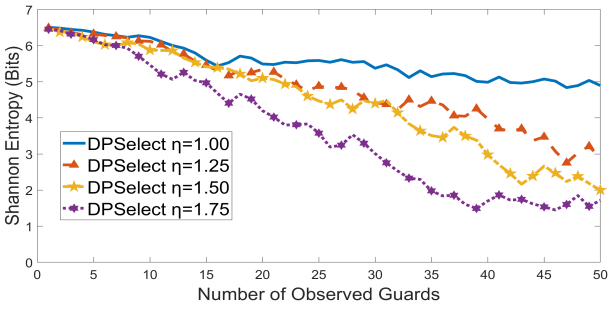

Fig. 25. η-value vs Minimum Shannon entropy of DPSelect after multiple guard relay selections in 100 simulations for clients in AS5432.

As seen from Fig. 25, the Shannon entropy for client AS5432 decreases over multiple guard relay selections as the η-value increases. Despite this, the Shannon entropy remains relatively high across all configurations. For instance, when $\eta = 1.75$ even after 50 guard relay selections, the worst-case Shannon entropy is still above 1-bit.

Min-Entropy: The min-entropy of a client I is calculated using Equation 5: $H_{Min}(I) = -\log_2 \max p_i$. We see the same relationship for min-entropy as we saw for Shannon entropy. As the η-value increases in Fig. 26, the lowest min-entropy observed decreases as well. However, only after 25 guard selections does any configuration reach 1-bit in the worst-case.

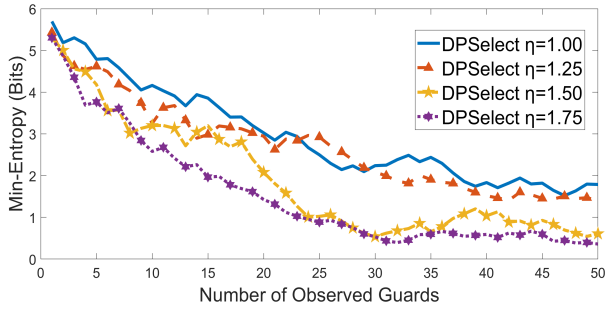

Fig. 26. η-value vs Lowest min-entropy of DPSelect after multiple guard relay selections in 100 simulations for clients in AS5432.

Guessing Entropy: The guessing entropy for a client I is calculated using Equation 6: $H_{Guessing}(I) = \sum_i i \cdot p_i$. As with the for min-entropy and guessing entropy, all potential η-DPSelect algorithms do better than Counter-RAPTOR. However, in Fig. 27, we see sharp degradation of the smallest guessing entropy as the number of observed guards increases.

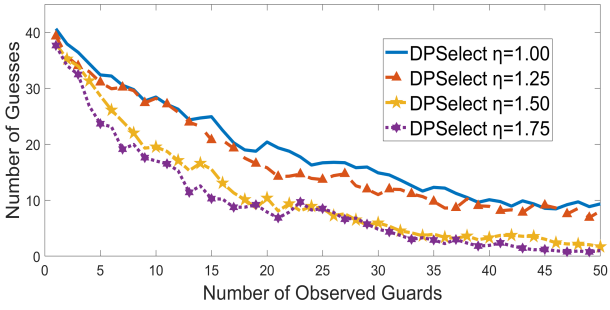

Fig. 27. η-value vs Minimum Guessing entropy of DPSelect after multiple guard relay selections in 100 simulations for clients in AS5432.

B.1.2 Client Anonymity

For the anonymity bounds [2], we see similar results to the anonymity results in Fig. 17 in Section 5. Increasing the value of η does not severely loosen the anonymity bounds. In Fig. 28 even though the anonymity bounds increase as the η-value increase, the sender anonymity never exceeds Vanilla Tor's sender anonymity of 0.1 (see Fig. 17). This still follows intuition because DPSelect redistributes guard relay selection choices amongst relays that both high bandwidth and those that have high resilience.

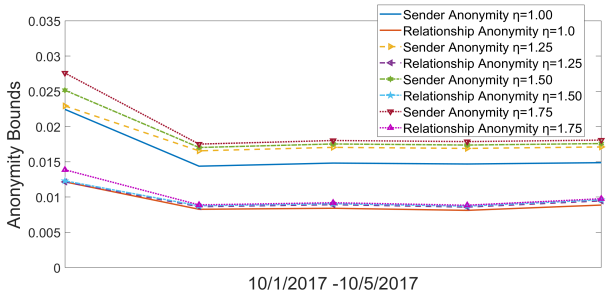

Fig. 28. MATor Anonymity Bound 10/1/2017-10/5/2017

B.1.3 Resilience to BGP Hijack Attacks

For BGP hijack attacks, increasing the η-value results in an increase in the overall resilience as seen in Fig. 29. Increasing the η-value causes the higher resilient guard relays to have a larger weight in the quality function.

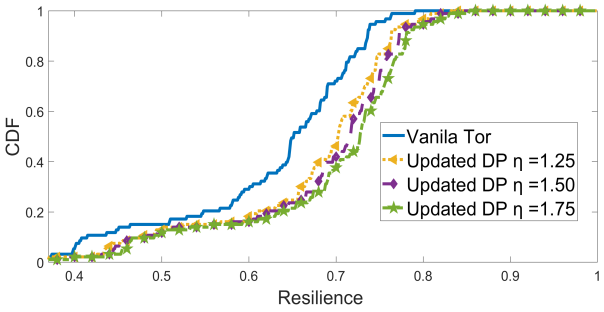

Fig. 29. CDF of the probability of being resilient to BGP hijack attacks for the Updated DP algorithm for different η-values.

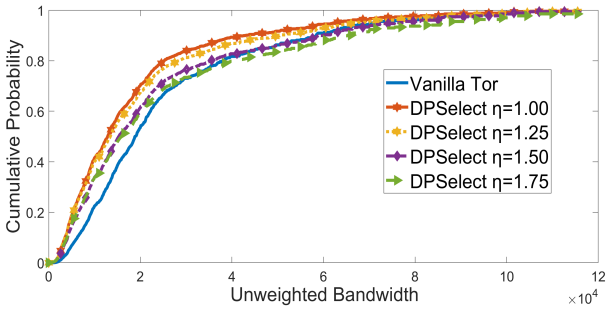

Fig. 31. CDF of the probability of being resilient to BGP hijack attacks for the Updated DP algorithm for different η-values.

B.2 Varying η-Value: Performance Evaluation

Here we evaluate and compare the performance of DPSelect for different η-values. We look at performance from three different perspectives:

1. **Bandwidth:** The average bandwidth of a selected guard relay across the top 93 Tor client ASes.
2. **Load Balancing:** Load balancing amongst guard relays.
3. **Shadow Emulation:** Performance evaluation using Shadow Tor emulator [18].

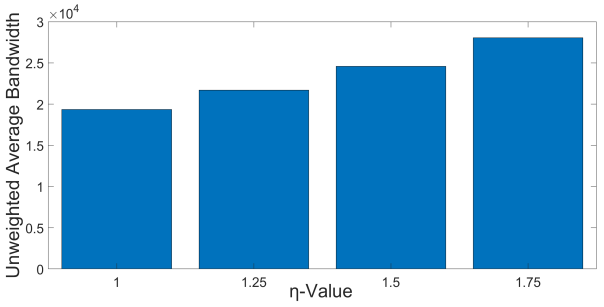

Fig. 30. Average bandwidth of guard relay choice for the Updated DPSelect algorithm for different η-values.

B.2.1 Bandwidth:

In terms of average bandwidth, as the η-value increases, the average bandwidth also increases. We see that when $\eta = 1.75$, the average bandwidth of a guard exceeds the average bandwidth of Vanilla Tor. This means that a larger share of higher bandwidth relays would receive traffic.

B.2.2 Load Balancing:

Choosing a high η-value restricts the set amongst which Tor clients choose their guards. As the η-value increases, the load is shifted more to the high bandwidth relays as seen in Fig. 30. For the η-values that we are examining, the load is still more balanced amongst low-bandwidth relays, but after $\eta = 1.5$, DPSelect begins to shift more drastically to higher bandwidth relays.

B.2.3 Shadow Emulation:

In order to do a more thorough investigation of how η-value affects performance, we run the implemented version of DPSelect with η-values from 1.25 to 1.75 in Shadow (the $\eta = 1.0$ gave us similar results to these values so we choose not to show it here). As in Section 5, we use the default Shadow configurations [18]. We also show $\eta = 5.0$ to illustrate how large η-values can degrade performance.

As can be seen in Figs. 29-32, as the η-value increases, the performance continues to degrade in the generated Tor network. In terms of average throughput in sending and receiving across all nodes in the network, we can see a downward trend in as the eta value increases. However, this decrease is only slight and only for $\eta = 5.0$ do we see a large decrease in performance.

Figs. 29-32 confirm our intuition of the higher bandwidth relays becoming congested as more Tor clients have a higher probability of selecting them. Thus, even though higher η-value result in better resilience to BGP hijack attacks, they do lead to worse performance. However, the decrease in performance in the generated network, while noticeable, is not large. Further, $\eta = 1.75$-DPSelect had a slightly better download time that either $\eta = 1.50$-DPSelect and $\eta = 1.25$-DPSelect.

B.3 Summary of Varying η-value

We have shown that when η-value increases, we observe different the security and privacy characteristics for DPSelect. Across Shannon entropy, min-entropy and guessing entropy, we see worse behavior as η-value increases.

However, increasing the η-value would allow the average resilience to increase and the average bandwidth to be closer to with Vanilla Tor, however. Given the

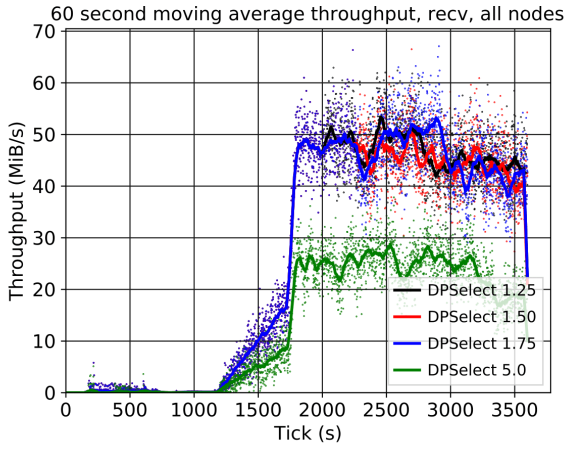

Fig. 32. 60 second average receiving throughput for all nodes.

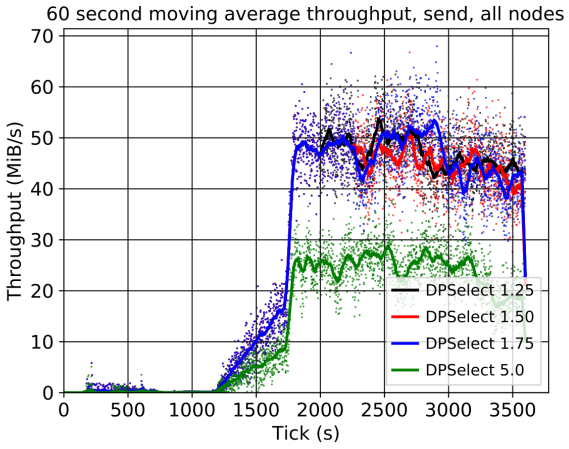

Fig. 33. 60 second average throughput for all nodes.

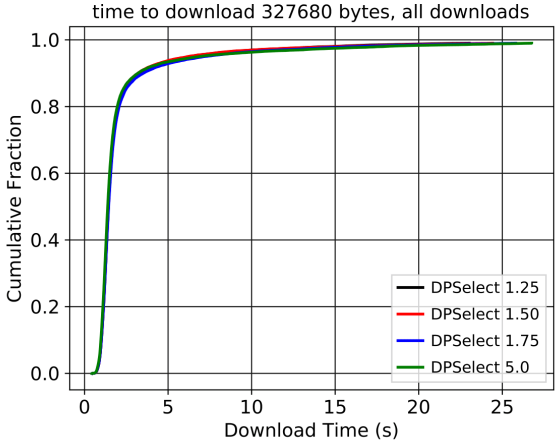

Fig. 34. Download time for 320KB data.

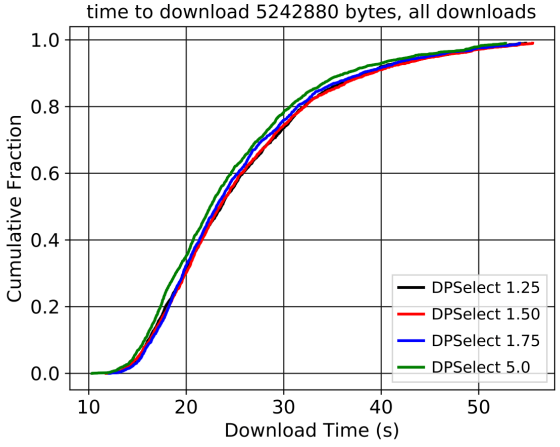

Fig. 35. Download time for 5MB data.

trade-offs picking an η-value in the range of 1.00 to 1.75 depends on the timescale of the attack against which we are protecting. More analysis needs to be done in order to determine realistic timescales for the amount of guards a particular Tor client observes.