DigesTor: Comparing Passive Traffic Analysis Attacks on Tor

Katharina Kohls[1](✉) and Christina Pöpper[2]

[1] Ruhr-University Bochum, Bochum, Germany
katharina.kohls@rub.de
[2] New York University Abu Dhabi, Abu Dhabi, United Arab Emirates
christina.poepper@nyu.edu

Abstract. The Tor anonymity network represents a rewarding target for de-anonymization attacks, in particular by large organizations and governments. Tor is vulnerable to confirmation attacks, in which powerful adversaries compromise user anonymity by correlating transmissions between entry and exit nodes. As the experimental evaluation of such attacks is challenging, a fair comparison of passive traffic analysis techniques is hardly possible. In this work, we provide a first comparative evaluation of confirmation attacks and assess their impact on the real world. For this purpose, we release *DigesTor*, an analysis framework that delivers a foundation for comparability to support future research in this context. The framework runs a virtual private Tor network to generate traffic for representative scenarios, on which arbitrary attacks can be evaluated. Our results show the effects of recent and novel attack techniques and we demonstrate the capabilities of *DigesTor* using the example of mixing as a countermeasure against traffic analysis attacks.

Keywords: Tor, Traffic Analysis, Confirmation Attack, Mixing

1 Introduction

With more than 2 million daily users [29] and 7000 active relays, Tor [28] is the most prominent example of an anonymity system that took the step from a scientific concept into the real world. Tor protects user privacy on the Internet by separating the origin of a connection from the requested services using onion-encrypted circuits. This mechanism cannot differentiate between benign use cases like censorship circumvention and malicious or illegal activities, but it protects the identities of both groups equally. While legal authorities are motivated in revealing identities to prosecute criminal behavior, censoring authorities can apply the same techniques to identify the origin of unwanted contents or links and thereby maintain control over the dissemination of information.

The existence of successful de-anonymization attacks against Tor is tremendously impactful because of its broad use. As a consequence, many classes of attacks have been introduced that attempt to reveal sensitive information about entities in the network [1, 8, 19]. These academic approaches are an essential building block for improving Tor, by more clearly defining the threat model it must address. At the same time, we may wonder if documented research attacks fully encapsulate the threat model experienced by Tor in practice. Scientific attacks *can* pose a serious threat and affect millions of users, but are driven by a focus on novelty rather than realism. This leads to a fundamental challenge of estimating an attack's real-world impact.

Our focus in this work is on passive traffic analysis attacks. These attacks are a current concern to the Tor community, in which user anonymity is compromised by an adversary correlating transmissions at the entry and exit of the circuit [20, 26]. Recent work [1,2,14,16] has demonstrated that an autonomous system (AS)-level adversary can successfully conduct confirmation attacks, correlating the characteristics of transmitted data to identify connections within the network.

The weakness to correlation is aggravated by routing attacks and nation-state adversaries with capabilities to surveil substantial fractions of the network. Border Gateway Protocol (BGP) attacks like RAPTOR [27] can increase the efficacy of confirmation attacks by directing traffic through an adversarial AS. This allows adversaries to have a near-total view of the network, a threat model not addressed by Tor. Mitigating traffic confirmation attacks, in particular against a *global adversary*, remains an open research problem [30].

Tor threat research is lacking comparable evaluation methodologies. Instead, analyses have been very divergent, ranging from theoretical models on the basis of statistical assumptions [3, 4], to approximate simulation systems [11], to experiments on the live Tor network [32, 33]. Theoretical models provide upper and lower bounds, but are limited by the assumptions made. Simulated systems can incorporate more real-world characteristics and often analyze network characteristics at a realistic scale, but only approximate certain parameters like the dynamics of an underlying network. The complexity of real-world network conditions makes it impossible to define holistic models that cover all potential cases, a fact that only allows for an estimation of effects on theoretical models and simulations. In contrast, experiments on the live Tor system demonstrate realistic conditions. However, especially in the context of traffic analysis attacks, work on the live network puts users at risk and is ethically discouraged [31].

Beyond the individual strengths of each of these methods, their *diversity* has led to a fundamental drawback: it is difficult to compare different attacks or understand their combined impact. This lack of comparability hinders the ability to understand existing attack vectors and progress defensive research in response.

We introduce *DigesTor* to address this fundamental shortcoming. *DigesTor* is an evaluation framework that guarantees comparability for recent, current, and future passive traffic analysis attacks, combining the strengths of simulated and real-world evaluation. The framework runs a virtual private Tor network to generate traffic for representative scenarios on which arbitrary attacks can be evaluated. The network uses virtual machines with individual CPU cores for each node and transmissions of realistic traffic through the actual network stack. Intermediate links simulate realistic network conditions using traffic shaping with parameters from empirical measurements in the live Tor network. This experimental setup increases realism over artificial traffic generation in simulated environments [11], can provide realistic link models, and satisfies the ethical guidelines for Tor research.

DigesTor includes a suite of state-of-the-art attack techniques that we evaluate using our framework. As a starting point for future work, this analysis provides a first performance comparison of existing attacks for their de-anonymization capabilities. Also, we demonstrate *DigesTor* by evaluating the use of delays as a potential countermeasure. The results of our attacks are summarized on `https://digestor.selfip.org` to demonstrate the features of our framework.

In short, our major contributions are:

- We release *DigesTor*, a comprehensive evaluation framework for passive traffic analysis attacks on Tor. This framework provides a basis to enable a fair comparison of existing and future attacks, is made publicly available, and includes an extensive corpus of transmission traces.
- We demonstrate the usefulness of *DigesTor* to evaluate the performance of state-of-the-art attack techniques. This leads to a first empirical overview of attack performance for different exemplary use cases and is a starting point for the development of future techniques.
- We use *DigesTor* to analyze low-latency mixing as a potential countermeasure to passive traffic analysis attacks. Results show that mixing, in fact, can counter confirmation attacks at a limited performance overhead only.

2 Traffic Analysis in Tor

Tor is a circuit-based transmission system that selects paths over network relays to form circuits. Usually, a circuit consists of one entry node, one middle node, and one exit node. Through successive layers of encryption to each relay, Tor separates the source of data from its destination, offering anonymity. We survey the attacks known to exist against Tor as follows and discuss two empirical adversary models.

2.1 Traffic Analysis Attacks

Tor defends against a set of known attack vectors, for instance, by ensuring unlinkable byte patterns through layered onion encryption. However, Tor ensures low-latency transmissions that trade performance for vulnerability against traffic analysis attacks. We introduce different classes of such attacks as follows.

In general, traffic analysis attacks exploit side channel information of encrypted transmissions through the network. This allows an adversary to monitor activities in the underlying network and reveal related connections. We distinguish the attack type, if an attack is (A/P) active (●) or passive (○), adversary model (Adv.) (◐: partial adversary, ●: global adversary), the evaluation setup

(●: evaluated in live Tor, ◐: reduced private network model, ○: theoretical model), the consideration of background noise (●: real noise, ◐: empirical noise, ○: statistical noise), and the consideration of different application types (App.) (●: yes, ○: no). Furthermore, we document whether a traffic metadata feature was used and define an attack metric (`Corr`: Correlation, `MI`: Mutual Information, `Enc`: Encoding, `Cell`: Cell Manipulation, `Blend`: Blending, `Stat`: Statistical Analysis.)

Table 1. Overview of end-to-end confirmation attack classes. In the Traffic Analysis Framework, we focus on passive attacks and flow comparison attacks.

Attack	**Ref.**	**A/P**	**Adv.**	**Setup**	**Noise**	**App.**	**Feature**	**Metric**
Flow Comp.	[15,26]	○	●	◐	◐	○	`iat`	`Corr`
	[14,35]	○	●	◐	◐	○	`iat`	`MI`
IXP Samples	[21]	○	◐	●	●	●	`iat`	`Stat`
Disclosure	[3,4,13,18]	○	●	○	○	○	-	`Stat`
Watermarking	[1,7,8,32]	●	◐	●	●	●	`iat`	`Corr`
Coding	[16,17,24,34]	●	◐	●	●	○	-	`Enc`
Protocol	[6,9]	●	◐	●	●	○	-	`Cell`
n-1	[5,23,25]	●	●	○	○	○	-	`Blend`

Passive Flow Comparison. A passive adversary monitors traffic at strategic points in the network and tries to detect related streams to de-anonymize users. This is accomplished through similarity/distance metrics that reveal relations between traffic measured at the source (client) and destination (server) of a connection. For this comparison, features like the timing between arriving packets [14, 26] are derived from unencrypted packet headers or transmission dynamics.

Further Attack Classes. In the literature we find further classes of passive attacks, e. g., disclosure attacks [4] or IXP (Internet exchange point) sampling [21] use statistical methods to compute the probability of two streams being related. An active adversary extends the scope of passive attacks by interference with the traffic stream, e. g., for injecting watermarks [1] or specific codes [16] that help to distinguish individual streams.

Even though the above attack landscape is motivated by the shared goal of learning sensitive information about anonymity systems and their users, we see a high diversity in their evaluation approaches. One example for this are statistical attacks, where we face evaluation results either from the live network [21] or a fully theoretical setup [4]. *DigesTor* overcomes this diversity by providing a consistent evaluation framework for passive flow comparison attacks. In particular, the use of *DigesTor* enables us to analyze the *technical* limitations of existing attacks. Furthermore, we introduce two empirical adversary models as follows.

2.2 Empirical Adversary

Besides their technical limitations, the success of traffic analysis attacks further depends on the adversarial network coverage, i. e., the probability of monitoring the *correct* Tor relays increases with higher coverage [27]. In a worst-case scenario, a global adversary has access to all traffic in the network. While this is a highly restrictive assumption and not considered in Tor's original attacker model, recent empirical studies reveal the potential threat of colluding and nation-state adversaries that achieve a significant coverage of the network. We derive two empirical attacker models from this:

1. *Partial Passive Attacker*: Approximately 40 % of Tor circuits are vulnerable to confirmation attacks by a single malicious AS [22]. This threat represents the view of an "average" adversary—or the impact of compromising an individual AS at the core of the Internet.
2. *Strong Partial Passive Attacker*: When ASes are considered on the state level, an adversarial nation could potentially compel multiple ASes within its governance to collude in correlation. Such an adversary could observe as many as 85 % of circuits [27].

However, for our experiments (see Section 5), we consider the *global passive* adversary as the upper bound. This encompasses weaker models, where a decreased network coverage limits the success probability of an attack (see Section 4.4).

3 *DigesTor* Framework

DigesTor is an open source analysis framework that provides comparability for the evaluation of passive traffic analysis attacks. We provide a high-level overview and introduce its evaluation set in the following.

3.1 System Components

DigesTor provides two core features: a Traffic Analysis Framework and a Virtual Private Tor Network. The *Traffic Analysis Framework* applies a set of attack techniques from related work to traces of our experimental network and outputs a performance assessment regarding the success of existing confirmation attacks. The framework covers five comparison metrics, which estimate the similarity or distance between observations in the network, i. e., pairs of client and server traces.

The *Virtual Private Tor Network* is used to generate network traffic that corresponds to typical use case scenarios. The traces are the monitored traffic streams an adversary would gather in a confirmation attack and are thus used as an input to generic passive end-to-end confirmation attacks. We use a virtualized private network for two main reasons. First, isolating the setup protects users of the live Tor network and ensures we do not violate the existing ethical guidelines for Tor research [31]. Second, the technical characteristics of a virtual setup provide significant advantages compared to a simulated setup. Using virtual machines for all nodes in the network, we utilize the actual protocol stack and transmit realistic application data. To improve the realism of our private network, we use empirical link models to imitate transmission delays monitored in the live Tor network.

3.2 Traffic Analysis Framework

In the following, we detail the traffic analysis component of *DigesTor*. Recent work suggests two types of metrics for flow comparison attacks. Correlation-based [15, 26] attacks compute the similarity in monitored traffic and identify relations between streams using the inter-arrival times, i. e., time periods between packets. Mutual information [14, 35] is a measure of the dependence of two streams and estimates similarity based on the entropy of observed pairs. Again, inter-arrival times are mentioned as a traffic feature for this type of attack.

From the current state of passive end-to-end confirmation attacks, we adopt the Pearson correlation coefficient (`P`) and Mutual Information (`MI`). We extend this by the Root-Mean-Square Error (`RMSE`) as a measure of distance between two observations, and a scalar comparison (`SC`) of features, in which we compare the sum of a metadata vector. Moreover, we sample an optional preprocessing step with the combination of the principal component analysis and Pearson correlation (`PCA-P`).

Eventually, we measure the success of an attack through the number of correctly guessed client/server connections, defined as *success rate*, and compare its improvement over random guessing, defined as ΔRG. The success rate describes the relative number of correct guesses in a setup, whereas the ΔRG indicates the strength of an attack. Furthermore, we use the area under the curve (AUC) for CDFs (cumulative distribution function) that summarize the results for combinations of multiple scenario setups. The AUC is a measure of the robustness of a successful attack, i. e., a smaller AUC indicates *higher* success rates.

3.3 Helpers

Besides the core components of *DigesTor*, we utilize a parser for transforming raw traces of network traffic to aggregated metadata vectors. More precisely, we extract a set of five features f_i: ($f_1 = \texttt{cnt}$) packet counts, ($f_2 = \texttt{iat}$) inter-arrival-timing, ($f_3 = \texttt{len}$) packet length, ($f_4 = \texttt{ttl}$) time to live, and ($f_5 = \texttt{wis}$) TCP window size. This metadata can be read from the header information of a TCP/IP packet ($\texttt{len}, \texttt{ttl}, \texttt{wis}$) or derived from packet occurrences ($\texttt{cnt}, \texttt{iat}$).

Using a window-based aggregation [15, 26], an average of all packets falling into one window is collected, e. g., for a measurement of 10 s and a window length of 0.1 s, we aggregate data in 100 equidistant windows. This results in time vectors of metadata information $(f_{i,1}, f_{i,2}, \cdots, f_{i,n})$ with features f_i over n time windows.

This feature set is parsed for each connection and filtered in the downlink direction (data flow from server to client). The feature set is non-exhaustive but extends the standard features in the literature (packet counts, inter-arrival times) with three more characteristics (packet length, window size, time to live) whose relevance will be part of the experimental analysis in Section 5.

4 Experimental Setup

In our experiments, we perform a comparative performance evaluation of attack metrics and demonstrate *DigesTor* by analyzing mixing as a potential countermeasure against traffic analysis attacks. We introduce the experimental setup, define the analyzed use case scenarios, and discuss the influence of Tor's network infrastructure as follows.

4.1 Technical Specification

Our experimental network (cf. Fig. 1) is defined by the different node types, i. e., clients, servers, and Tor relays, and by the topology that connects them.

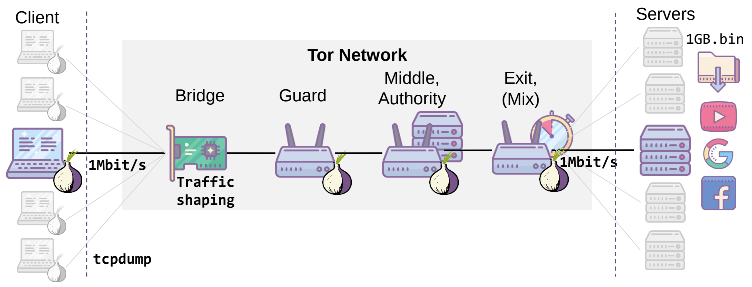


Fig. 1. Clients connect to servers through circuits of three relays. The bridge applies empirical traffic shaping for each client connection individually. Servers provide random binary files for downloads or proxy web requests [10].

Nodes. Entities in the network are configured to serve as i) clients that make requests through Tor, ii) servers that provide requested data, and iii) relay nodes that build the private Tor network. Each client connects to a predefined server and follows a use case scenario which includes either download requests via the *cURL* library or website browsing using the browser automation framework *Selenium* and a headless browser (developed as part of *Mozilla Firefox*). Requests are made through the SOCKS5 proxy at port 9050. They are synchronized via NTP for all clients, i. e., experiments start and end at the same time. The server nodes provide file downloads over HTTP at port 80 and reverse proxy requests to a set of Alexa Top 50 websites at port 80 and 443. We use three relay nodes of which one is configured as guard, one as middle and authority, and one as exit relay. The relay, authority, and client nodes run Tor version `0.2.9.8`.

Network. We use an empirical link model for the downlink connection of all clients. The link model adds per-packet delays drawn from measurements of arbitrary circuits in the live Tor network, which are individually assigned for each connection. This traffic shaping is accomplished by a bridge interface, where each client connection samples from an individual delay distribution. For the network topology either a directed setup, using 1:1 connections between n clients and n servers, or a grouped setup, using n:2 connections between n clients and two servers, is used. The number of relays is fixed to three.

Hardware. The VMs run in a cloud space hosted in one central location, each node is assigned a distinct CPU core. The full setup can utilize up to 63 cores, 132 GB of RAM, and 504 GB of disk space. We capture the traffic of all client and server nodes using `tcpdump`. Raw network traces are gathered on one central file server for further processing outside the Tor network environment and therefore do not interfere with the performance of network nodes.

4.2 Scenarios

We test individual *topologies* of 2 to 30 clients to 2 to 30 servers in a *Directed* and of 2 to 30 clients to two servers in a *Grouped* setup. Furthermore, we distinguish three individual *application* models:

- **Static download.** The user requests a file from the server via *cURL* and permanently loads it during the entire duration of the measurement.
- **Random download.** Each user requests a file from the server via *cURL*, whereas on/off periods for the downloads are randomized for the entire duration of the measurement. Off periods are uniformly distributed between 2 s to 10 s, on periods are uniformly distributed between 2 s to 5 s.
- **Browsing.** From the Alexa Top 50 web pages, each client requests a random set of sites using a scripted headless browser. Between site requests, clients wait for a random period with a uniform distribution between 2 s to 5 s before the next request is sent.

We emphasize that the randomization of on/off periods can influence the results, as a higher variance in the duration of off periods helps to distinguish individual transmissions. Consequently, our results can only represent the parameter choices made above. We discuss the definition of more sophisticated use case scenarios in Section 6.

4.3 Comparison of Attack Metrics

In the following, we apply the Traffic Analysis Framework (combinations of features `cnt,iat,len,ttl,wis` and metrics `P,MI,RMSE,SC,PCA-P`) to all combinations (directed, grouped; static, random, browsing) and an increasing number of clients $n = 2$ to 30; each experiment is repeated for five random repetitions.

We compute the general attack success (AS: how many connections were guessed correctly?), the improvement over random guessing (ΔRG: how much better was the attack compared to an uneducated guess?), and the area under the curve (AUC: how convincing and robust was a result?) of the cumulative distribution function (CDF) of results.

4.4 Tor Network Infrastructure

While our experimental setup covers the *technical* comparison of attack metrics and traffic features, we are further interested how Tor's network infrastructure influences the *organizational* aspects of an attack. Therefore, we discuss the scalability of our setup and the relay selection process as a preliminary step to the performance comparison in Section 5.

Scalability In the setups we demonstrate, clients run at a maximum rate of 1 Mbit/s. For the described *Grouped* and *Directed* scenarios, this translates into a throughput of 30 Mbit/s passing through each of the relays. This scale places these relays within the top 10 % of active Tor relays by bandwidth. Experiments with fewer active clients would approximate the traffic of less active relays, with approximately $\frac{2}{3}$ of relays transmitting at least 4 Mbit/s of traffic, the level of traffic we simulate in our smallest experiments. We do not model the number of active connections experienced by Tor relays. While we can expect a total of 500,000 active clients at any given point [12], it is less clear how those clients are distributed across relays and bridges. However, the median relay will have less than 50 active clients regardless of the distribution. With up to 30 parallel connections our network setup achieves a similar relay workload.

Relay Popularity Tor's network infrastructure is skewed towards the countries where we find the most Tor supporters, e. g., Germany (19.4 %), the US (18.7 %), and France (14.2 %) maintain more than half of the entire network. Furthermore, higher bandwidth relays are preferred in the circuit buildup procedure. An attacker can benefit from these characteristics and focus on frequently used nodes, e. g., it is possible to cover 75 % of all *selected* exit relays by monitoring approximately 26 % of nodes (cf. Fig. 2). This situation supports the empirical adversary models (Section 2) and is incorporated by the attack setup of *DigesTor*.

5 Evaluation

We use the above experimental setup of *DigesTor* for a first comparative analysis to (i) derive the best performing metric and feature combinations for each setup, compare the characteristics of different (ii) topologies and application types, and (iii) analyze mixing as one possible countermeasure against traffic confirmation. Finally, we (iv) give an overview of the takeaway messages of our evaluation.

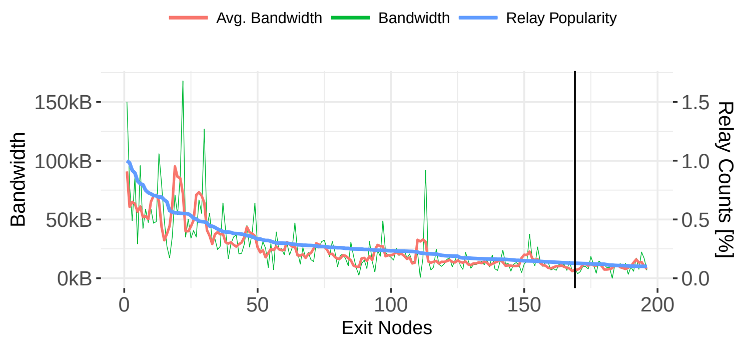


Fig. 2. Distribution of Exit relay popularity and respective advertised bandwidth, measured for a total of 100,000 Tor standard circuits.

5.1 Metrics and Features

As initial research question, we address the performance comparison of attack metrics and metadata features. Beginning with the overall *global* performance, we get a first impression of the impact of confirmation attacks in generic scenarios. We continue with an analysis of *individual* combinations of metrics and features for all scenarios.

Global Performance. In our first evaluation step, we identify the overall best-performing metrics and features for a combination of all scenario setups. Figure 3 summarizes the attack success, i. e., the relative number of successful connection identifications, for all traces in the *DigesTor* corpus. Each box represents the full performance range of a metric/feature, whereas we focus on the comparison median (horizontal bar) results. We see that Mutual information (`MI`) provides the best overall result (median=0.48) in a global comparison. This result summarizes the attack success for all combinations of `MI` with the given traffic features and applies for all scenarios introduced in Section 4.2. In the comparison of metadata features the time to live field (ttl) performs best (median=0.44).

Individual Performance Figure 4 highlights the performance of all individual combinations of metrics and features. Darker tiles in the heat map indicate a higher attack success at a specific experimental setup. Table 2 summarizes these results and provides an overview of the best performing metric and feature combinations for individual setups. We see that (`MI,iat`) performs best in a global comparison, i. e., it is the most robust combination while performing 23 % better than random guessing. Overall, `iat` is the most reliable metadata feature for most scenarios, whereas we see varying metrics for individual setups.

What Metric and Feature Combination Performs Best? Without any prior knowledge of the use case and number of concurrent transmissions, `MI/ttl`

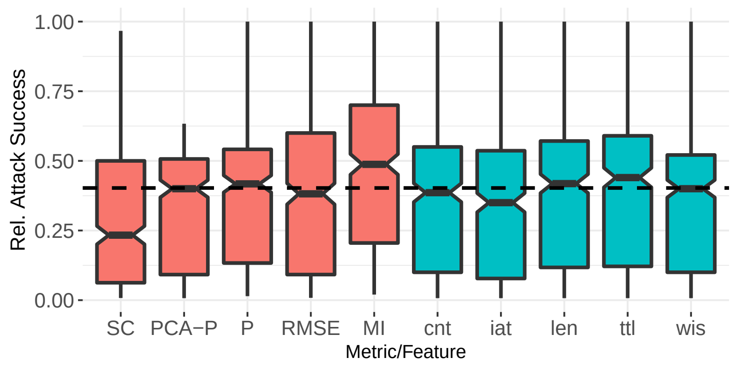


Fig. 3. Comparison of attack success for individual metrics (red) and features (blue) for all topologies and applications combined. Results show the median (horizontal bar in box) aggregated for 2 clients to 30 clients in comparison to the average success of random guessing (dashed line).

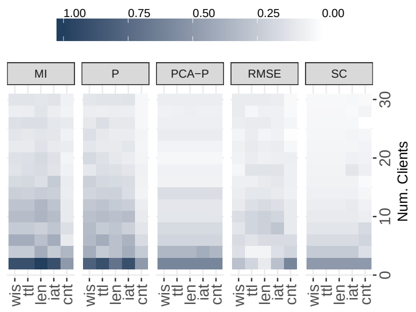
Table 2. Best performing metric and feature combinations. Results show the improvement over random guessing (Δ RG), global performance (AUC), and average success rate (AS) through all experiments.

Scenario	**Metric**	**Feature**	Δ **RG**	**AUC**	**AS**
Directed	`P`	`ttl`	35 %	0.72	0.49
Grouped	`MI`	`iat`	22 %	0.50	0.55
Random	`RMSE`	`cnt`	52 %	0.48	0.80
Static	`MI`	`iat`	16 %	0.65	0.46
Browsing	`SC`	`iat`	7.4 %	0.70	0.34
Global	`MI`	`iat`	23 %	0.61	0.52

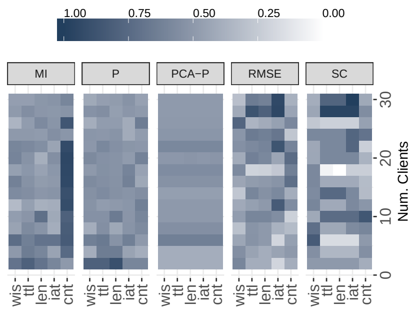
outperform an average random guessing attack. As soon as it is possible to adjust to a certain scenario, the targeted combination of a metric and feature helps to increase the improvement over random guessing.

5.2 Scenarios

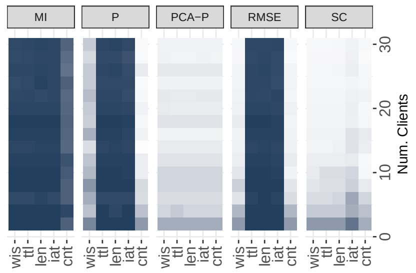
Different topologies have two characteristics that influence the success of an attack. First, grouped setups, where n clients connect to only 2 individual servers, induce more noise through concurrent transmissions for traffic that is captured at the server. Such noise complicates the application of comparison metrics and destroys connection-individual parameters. One example for this is the attack success for a *random* download in the directed (cf. Fig. 4(c)) and grouped (cf. Fig. 4(d)) topology. We see that it is possible to distinguish connections even for high user numbers in the directed setup (ΔRG=35 %), whereas we



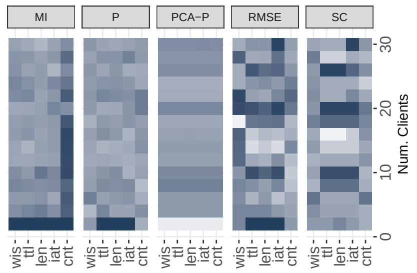
(a) Directed Static



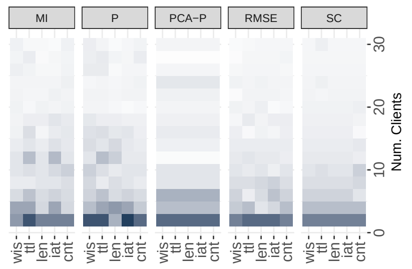
(b) Grouped Static



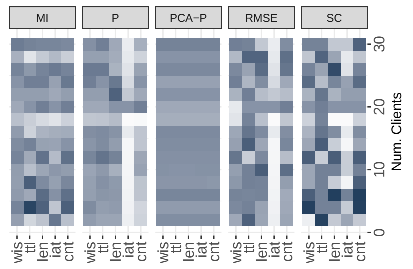
(c) Directed Random



(d) Grouped Random



(e) Directed Browsing



(f) Grouped Browsing

Fig. 4. Average performance of all metrics and features for both topologies and all application types. The heatmap indicates the relative attack success, ranging from 0, no success, lighter to 1, high success, darker.

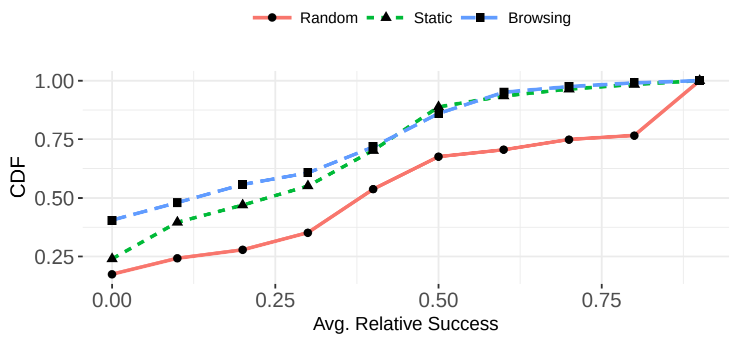


Fig. 5. Cumulative Distribution Function of average attack success for the comparison of three use case scenarios.

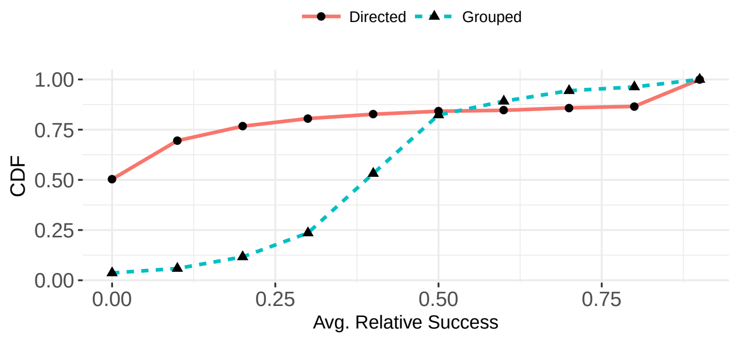


Fig. 6. Cumulative Distribution Function of average attack success for the comparison of directed and grouped network topologies.

lose too much information in the grouped experiments (ΔRG=22 %). Second, the number of candidates for guessing a connection is limited to two serves in the grouped setup. Consequently, we experiencemore stable results for grouped topologies (AUC=0.5) than in directed setups (AUC=0.72) with overall more connection candidates.

What Scenarios Favor Attacks? Guessing on fewer candidates (grouped topology) makes it easier to achieve positive success rates for an attack. At the same time, it becomes harder to distinguish individual traffic characteristics through simple comparison metrics. Our results show that random downloads, where a high amount of data is sent in individual patterns, provide the best improvement over an uneducated guess. In combination with a setup that reduces noise of concurrent transmissions, this leads to a successful attack even for higher user numbers. The same does not apply to user-individual browsing, where traffic patterns are unique but the amount of data sent is insufficient for distinguishing connections reliably.

5.3 Countermeasures.

We can counter traffic analysis by perturbing traffic features during the transmission process. One example for this is mixing [35], where intended delays for packets change the timing relations of a connection. As such countermeasures can decrease a system's performance, we analyze mixing concerning its protection capabilities and performance impairments.

Implementation. We implement a mix within the Tor code and deploy it in the exit relay of our experimental setup. The mix delays TLS records within Tor before they are emitted for further transmission; it uses a defined delay duration (time held back) and rate (relative amount affected). TLS records are, within a Tor relay, closest to the transport layer on which an adversary monitors connections. We, therefore, expect a maximum effect on traffic analysis attacks. In the following, we give an example for different mix delays (time added to sending of TLS records) and mix rates (a portion of records affected by mixing). The mix does not provide any differentiation of TLS records from different connections, e. g., mixing is applied to a fraction of *all* records in the relay.

Results. At a static mix rate of 20 % (directed network topology, static download application), we achieve an AUC in the range of 0.9 to 0.95 for delay durations between 10 ms to 1 ms, which represents at least 20 % improvement over the unmixed attack success (AUC=0.72). At the same time, we see that varying mix rates do not influence the attack success significantly.

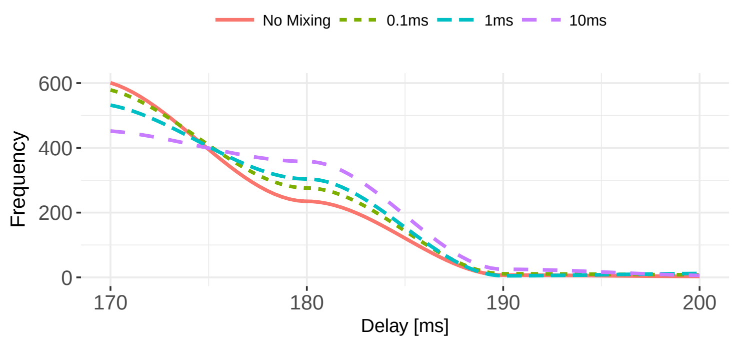


Fig. 7. Distribution of end-to-end delays measured in our experimental setup. Results show slightly increasing round trip time for mixed setups, where we tested a static mixing rate of 20 % and increasing mix delays.

Moreover, we analyze the end-to-end delays for increasing mix delays at a fixed rate of 20 %. Results show slightly increased delays for mixed connections, while the performance impairments are still in an acceptable range. **Does Mixing Counter Attacks?** Our results support the concept of mixing, whereas the delays can only protect a subset of metadata features. The achieved obfuscation is sufficient for casual scenarios at an acceptable performance overhead, but at this price cannot guarantee *perfect* traffic analysis resistance.

5.4 Overview of Results

We summarize the results of our experimental evaluation as follows.

1. **Metrics and Features Combined.** For all topologies and applications we found a metric and feature combination that outperformed random guessing (Table 2). These combinations do *not* focus on a single traffic feature, hence, an isolated obfuscation of metadata features cannot protect against traffic analysis in general.
2. **Topologies and Applications.** Even though we found topologies and applications that hinder an attack, the attack framework outperformed random guessing attacks by 26.48 % on average (individual scenarios) and 23 % in generic scenarios.
3. **Affordable Countermeasures.** We use the comparative evaluation of *DigesTor* to demonstrate low-latency mixing as a countermeasure to traffic analysis attacks. Such effects can be achieved at minimal additional delays of 1 ms, which renders this solution an actual option for the live system.

6 Discussion

After demonstrating the experimental benefits of our traffic analysis framework, we now introduce how *DigesTor* can be used to support future research and what limitations the system faces at the moment. Furthermore, we discuss the ethical guidelines for this work and the potential of mixing as a countermeasure.

6.1 Goals of *DigesTor*

The goal of our evaluation framework is to accelerate the deployment of new defenses. To achieve this, we must provide a set of conditions which appropriately represent Tor's infrastructure, but also operate at sufficient scale to approximate the parameters of the real network.

How to use *DigesTor*? The results of this work provide a first comparative overview of attack metrics and metadata features. Our work supports future research as follows.

- **Trace Corpus.** Our trace corpus represents standard topologies and application types and can be used to evaluate generic passive attacks without harming users of the live network. Furthermore, this once more supports the comparability of results.

- **Attacks.** The traffic analysis framework already provides a representative set of metrics and can be extended further by new attack metrics and meta-data features. This allows comparing new approaches with the success of existing work.
- **Defenses.** Following the example of mixing as a countermeasure, future defensive research can use the performance comparison to assess the effects of novel countermeasures.

Limitations. For the use case scenarios, we approximate real user behavior by simple models, e. g., through randomized web requests to a restricted set of sites or random download patterns. This does not represent the user behavior that defines the traffic patterns in a real-world scenario. In end-to-end confirmation attacks, a matching between client and server traces is the primary interest. Adding user models to the experimental setup in a future revision of *DigesTor* helps to create more realistic scenarios, but is not crucial for the technical evaluation of attacks.

6.2 Ethics

In compliance with the Tor Ethical Research Guidelines [31], we designed this work in a way that does not harm users of the live network. We emphasize that especially the experimental evaluation of traffic analysis attacks can cause damage to real-world users and should always be conducted in a controlled environment. In turn, this applies to the analysis of countermeasure implementations whose security yet has to be proven.

6.3 Mix Countermeasure

Our TLS mix concept is implemented at exit nodes and can support a slow rollout over the existing network. Mixing of TLS records means there cannot be mixed and unmixed connections at the same time in one relay, reducing the unmixed bandwidth for the sake of increased security. However, not all nodes in the network must provide mixing, as a small fraction is sufficient to introduce uncertainty for an adversary across many active circuits. Along with the dynamic adaption of mix parameters, this makes the mix concept flexible: instead of using fixed setups, mix parameters can be coupled with monitoring the current network status and load.

7 Conclusion

DigesTor is an appeal to comparability in security research on Tor. The attack landscape of current research offers various classes of offensive work that *might or might not* pose a threat to the live Tor network. With *DigesTor* we share two core features: We generated a first traffic analysis corpus of this kind that we share to support the comparability of future research. The second core feature is the Traffic Analysis Framework, which applies a set of recent attack techniques for comparative performance analysis. To demonstrate the benefits of *DigesTor*, we analyze mixing as a potential countermeasure against passive traffic analysis attacks. Our results indicate that mixing, in fact, hinders the success of otherwise successful confirmation attacks.

Acknowledgments

This work was supported in part by Intel (ICRI-CARS) and the German Research Foundation (DFG) Research Training Group GRK 1817/1. We would like to thank Maximilian Golla for his support with the experimental setup.

References

1. A. Biryukov, I. Pustogarov, and R.-P. Weinmann, "Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization," in *Symposium on Security and Privacy*. IEEE, 2013, pp. 80–94.
2. S. Chakravarty, M. V. Barbera, G. Portokalidis, M. Polychronakis, and A. D. Keromytis, "On the Effectiveness of Traffic Analysis Against Anonymity Networks Using Flow Records," in *International Conference on Passive and Active Measurement*, ser. PAM '14. Los Angeles, CA, USA: Springer, Mar. 2014, pp. 247–257.
3. G. Danezis, "Statistical Disclosure Attacks," in *Security and Privacy in the Age of Uncertainty*. Springer, 2003, pp. 421–426.
4. G. Danezis, C. Diaz, and C. Troncoso, "Two-Sided Statistical Disclosure Attack," in *Workshop on Privacy Enhancing Technologies*, ser. PET '07. Ottawa, ON, Canada: Springer, Jun. 2007, pp. 30–44.
5. C. Diaz and B. Preneel, "Taxonomy of Mixes and Dummy Traffic," in *Information Security Management, Education and Privacy*. Springer, 2004, pp. 217–232.
6. X. Fu, Z. Ling, J. Luo, W. Yu, W. Jia, and W. Zhao, "One Cell is Enough to Break Tor's Anonymity," in *Proceedings of Black Hat Technical Security Conference*, 2009, pp. 578–589.
7. A. Houmansadr and N. Borisov, "SWIRL: A Scalable Watermark to Detect Correlated Network Flows," in *NDSS*, 2011.
8. ——, "The need for Flow Fingerprints to Link Correlated Network Flows," in *Privacy Enhancing Technologies Symposium*. Springer, 2013, pp. 205–224.
9. A. Houmansadr, C. Brubaker, and V. Shmatikov, "The Parrot is Dead: Observing Unobservable Network Communications," in *Symposium on Security and Privacy*. IEEE, 2013, pp. 65–79.
10. icons8. Figure Icons. Accessed: 2018-04-23. [Online]. Available: https://icons8.com
11. R. Jansen and N. Hopper, "Shadow: Running Tor in a Box for Accurate and Efficient Experimentation," in *Symposium on Network and Distributed System Security*, ser. NDSS '12. San Diego, CA, USA: Internet Society, Feb. 2012.
12. R. Jansen and A. Johnson, "Safely Measuring Tor," in *Conference on Computer and Communications Security*. ACM, 2016, pp. 1553–1567.
13. D. Kesdogan, D. Agrawal, and S. Penz, "Limits of Anonymity in Open Environments," in *Workshop on Information Hiding*. Springer, 2002, pp. 53–69.

14. A. Kwon, M. AlSabah, D. Lazar, M. Dacier, and S. Devadas, "Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services," in *USENIX Security Symposium*, 2015.
15. B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, "Timing Attacks in Low-Latency Mix Systems," in *International Conference on Financial Cryptography*. Springer, 2004, pp. 251–265.
16. Z. Ling, X. Fu, W. Jia, W. Yu, D. Xuan, and J. Luo, "Novel Packet Size-Based Covert Channel Attacks Against Anonymizer," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2411–2426, 2013.
17. Z. Ling, J. Luo, W. Yu, X. Fu, D. Xuan, and W. Jia, "A New Cell Counter Based Attack Against Tor," in *Conference on Computer and Communications Security*. ACM, 2009, pp. 578–589.
18. N. Mathewson and R. Dingledine, "Practical Traffic Analysis: Extending and Resisting Statistical Disclosure," in *Workshop on Privacy Enhancing Technologies*. Springer, 2004, pp. 17–34.
19. P. Mittal, A. Khurshid, J. Juen, M. Caesar, and N. Borisov, "Stealthy Traffic Analysis of Low-Latency Anonymous Communication Using Throughput Fingerprinting," in *Conference on Computer and Communications Security*, ser. CCS '11. Chicago, IL, USA: ACM, Oct. 2011, pp. 215–226.
20. S. J. Murdoch and G. Danezis, "Low-Cost Traffic Analysis of Tor," in *Symposium on Security and Privacy*, ser. SP '05. Oakland, CA, USA: IEEE, May 2005, pp. 183–195.
21. S. J. Murdoch and P. Zieliński, "Sampled Traffic Analysis by Internet-Exchange-Level Adversaries," in *Workshop on Privacy Enhancing Technologies*. Springer, 2007, pp. 167–183.
22. R. Nithyanand, O. Starov, A. Zair, P. Gill, and M. Schapira, "Measuring and Mitigating AS-level Adversaries Against Tor," in *Symposium on Network and Distributed System Security*, ser. NDSS '16. San Diego, CA, USA: Internet Society, Feb. 2016.
23. L. O'Connor, "On Blending Attacks for Mixes with Memory," in *Workshop on Information Hiding*. Springer, 2005, pp. 39–52.
24. H. Sengar, Z. Ren, H. Wang, D. Wijesekera, and S. Jajodia, "Tracking Skype Voip Calls Over the Internet," in *International Conference on Computer Communications*. IEEE, 2010, pp. 1–5.
25. A. Serjantov, R. Dingledine, and P. Syverson, "From a Trickle to a Flood: Active Attacks on Several Mix Types," in *Workshop on Information Hiding*. Springer, 2002, pp. 36–52.
26. V. Shmatikov and M.-H. Wang, "Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses," in *European Symposium on Research in Computer Security*, ser. ESORICS '06. Hamburg, Germany: Springer, Sep. 2006, pp. 18–33.
27. Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal, "RAPTOR: Routing Attacks on Privacy in Tor," in *USENIX Security Symposium*, ser. USENIX '16. Washington, DC, USA: USENIX, Aug. 2015, pp. 271–286.
28. The Tor Project. The Onion Router. Accessed: 2018-04-23. [Online]. Available: https://www.torproject.org
29. ——. Tor Metrics. Accessed: 2018-04-23. [Online]. Available: https://metrics.torproject.org
30. ——. (2014, Jul.) Tor Security Advisory: "Relay Early" Traffic Confirmation Attack. Accessed: 2018-04-23. [Online]. Available: https://blog.torproject.org/blog/tor-security-advisory-relay-early-traffic-confirmation-attack

31. ——. (2015, Nov.) Ethical Tor Research: Guidelines. Accessed: 2018-04-23. [Online]. Available: https://blog.torproject.org/blog/ethical-tor-research-guidelines
32. X. Wang, S. Chen, and S. Jajodia, "Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems," in *Symposium on Security and Privacy*. IEEE, 2007, pp. 116–130.
33. X. Wang and D. S. Reeves, "Robust Correlation of Encrypted Attack Traffic Through Stepping Stones by Manipulation of Interpacket Delays," in *Conference on Computer and Communications Security*. ACM, 2003, pp. 20–29.
34. W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "DSSS-Based Flow Marking Technique for Invisible Traceback," in *Symposium on Security and Privacy*. IEEE, 2007, pp. 18–32.
35. Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "On Flow Correlation Attacks and Countermeasures in Mix Networks," in *Workshop on Privacy Enhancing Technologies*. Springer, 2004, pp. 207–225.