On Evaluating Anonymity of Onion Routing

Alessandro Melloni[1][0000−0002−8851−6452], Martijn Stam[1][0000−0002−5319−4625], and Øyvind Ytrehus[1][0000−0001−5223−7577]

Simula UiB
Thormøhlensgate 53D
N-5006 Bergen, Norway.
{alessandro,martijn,oyvindy}@simula.no

Abstract. Anonymous communication networks (ACNs) aim to thwart an adversary, who controls or observes chunks of the communication network, from determining the respective identities of two communicating parties. We focus on low-latency ACNs such as Tor, which target a practical level of anonymity without incurring an unacceptable transmission delay.

While several definitions have been proposed to quantify the level of anonymity provided by high-latency, message-centric ACNs (such as mix-nets and DC-nets), this approach is less relevant to Tor, where user–destination pairs communicate over secure overlay circuits. Moreover, existing evaluation methods of traffic analysis attacks on Tor appear somewhat ad hoc and fragmented. We propose a fair evaluation framework for such attacks against onion routing systems by identifying and discussing the crucial components for evaluation, including how to consider various adversarial goals, how to factor in the adversarial ability to collect information relevant to the attack, and how these components combine to suitable metrics to quantify the adversary's success.

Keywords: Anonymity · Onion Routing · Tor · Traffic Analysis

1 Introduction

Anonymous communication networks (ACNs) enable users to communicate with each other while hiding as much as possible who said what to whom from an adversary. The focus of anonymity varies depending on what should remain hidden: for instance, who sent to whom versus who sent what. This variation is reflected in a plethora of precise formalizations of anonymity in the context of communication [2, 7, 27, 44], culminating in the recent 'race-car' hierarchy by Kuhn et al. [34, Fig. 3]. Two of the main concepts are unlinkability and unobservability, for instance sender–receiver unlinkability (an adversary cannot tell whether Anna is communicating with Bob or Dad) or sender unobservability (an adversary cannot tell whether Anna is communicating at all).

Of course, not all ACNs will, or even intend to, satisfy all possible notions. In fact, most ACNs belong to one of three main classes: DC nets (after Chaum's dining cryptographers [12]), mix-nets [11], and onion routing [29]. These classes

differ in their overhead, both in terms of bandwidth and latency. Typically, the less overhead and hence the more performant the ACN, the less formal guarantees one can hope to obtain [15]. Arguably, onion routing introduces the least inevitable overhead. It involves a user selecting a number of relays and encrypting a message in such a way that each of the relays peels of a layer of encryption until the final message is retrieved at the destination. Onion routing can be defined both in a public-key setting where each message can take its own route [9], or in a symmetric-key setting where a circuit is established on which a secure channel is overlayed [28, 46].

Tor [19] is an ACN of the latter type. It aims to improve online anonymity such that even someone monitoring parts of the network cannot easily tell which user is visiting which website, or, more generally, who is connected to whom. As mentioned above, users relay their data over multi-hop circuits, using encryption to hide routing data and thwart easy, content-based correlation of traffic going in and out of any given router. However, to keep overall network latency low, the timing of incoming and outgoing traffic is certainly correlated. Indeed, when Tor was conceived, it was accepted that ingress traffic traces collected at the guard can be linked with the corresponding egress traffic traces collected at the exit node. Thus, an adversary controlling both the guard node and the exit node of a circuit can use traffic analysis to deanonymize such a circuit, thereby linking its user to their destinations.

Yet, in reality the compromise is neither automatic nor complete, and different methods have been proposed to correlate ingress and egress traces [5]. A completely different kind of traffic analysis arises when an adversary fingerprints a list of websites and, with only access to a user's ingress trace, tries to determine which website the user is visiting [42]. In parallel, several defence mechanisms have been suggested to reduce the potency of these attacks (see e.g. [33] for an overview of both attacks and defences).

A natural question is how well these attacks, respectively defences, work: in other words, how to evaluate these attacks and defences. In addition to the two distinct threat models mentioned above, there are various goals that have been considered in the past, for instance determining whether a user is accessing a monitored or unknown website (the 'open' world) versus deciding which of a number of known websites a user is accessing (the 'closed' world). Moreover, different metrics are used to evaluate, depending on the scenario; for the open world's decisional problem one often sees precision and recall, whereas for the closed world's classification problem, accuracy is more common. In some cases, information-theoretic metrics have been used and advocated [4, 16, 49, 51].

What is lacking, though, is a common methodology or perhaps even language how best to evaluate and interpret attacks against and defences for Tor. The attacks themselves can often be regarded considerably less scenario-dependent than their evaluation indicates and for defences there is the legitimate question to what extent their evaluation should depend on current state-of-the-art of attacks. Finally, it is unclear how the attacks and their evaluations relate to the fine-grained formal definitions of anonymity mentioned above.

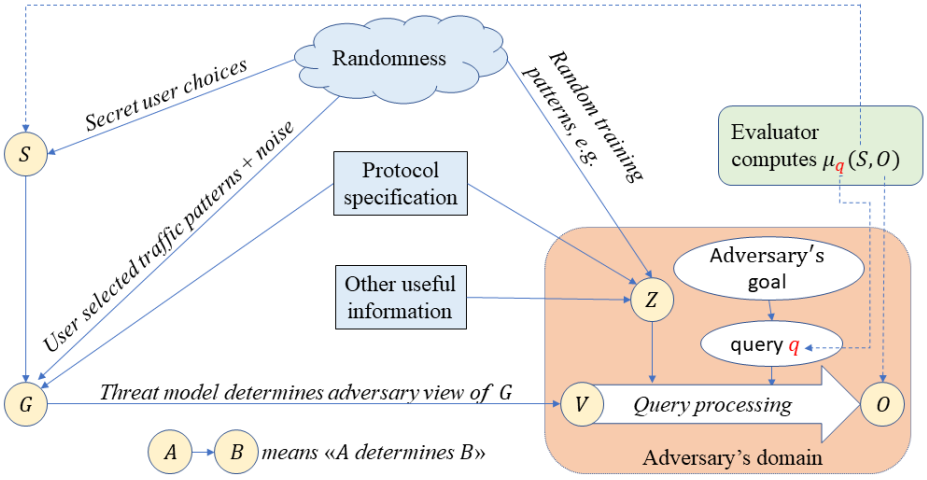


Fig. 1. View of the game and the random variables involved in it: The secret S underlies the game's state G of which an adversary can only observe V. Auxiliary information Z from earlier training is used to extract from V an answer O to the query $\mathtt{q}$.

Our contribution. We propose a framework for the evaluation of the anonymity offered by low-latency onion routing schemes such as the current Tor design. Our aim is to enable a fair comparison of various traffic analysis methods and related defences, by clarifying the possible threat models and providing a taxonomy of relevant security goals and appropriate metrics.

In Section 2, we cast the interplay between Tor and an adversary as a cryptographic game, identifying relevant random variables to express success of an adversary as a population parameter. Fig. 1 gives a very high level overview. Even when we cannot hope to ever fully learn the real-life distribution of said variables, for evaluation, one can still set a suitable, hopefully representative distribution and run partially simulated experiments, substituting the true population parameter for a sample statistic on an approximate distribution.

As Tor is used by real people with legitimate privacy concerns, ethical evaluation invariably uses a partially simulated and scaled down version. In a simulation, the evaluator can run multiple experiments, each time knowing the ground truth of who is connected to whom. That sets evaluation apart from an actual adversary trying to learn what is happening during a single snapshot of Tor.

Of course, a real-life adversary will operate against real world Tor and thus the threat models, and possible goals an adversary may have, derive directly from considering the real world. Specifically, an attacker against Tor typically will be able to observe some of the traffic flowing through the network and possibly to manipulate said traffic. Exactly which traffic can be observed depends on the threat model, as we elaborate upon in Section 5. For instance, there is a difference between an adversary only observing traffic flowing through the proxy versus an adversary who has corrupted multiple guard and exit nodes.

Table 1. List of random variables in the framework.

Random variable	Description
S	The game's 'secret' mapping of users with destinations
G	The game's full view of the interaction
V	The adversary's limited view of the interaction
Z	Auxiliary information given to or obtained by the adversary
O	The adversary's output, expressing belief about part of the secret

An adversary will use its observations to deduce information about who is connected to whom. There are various ways one can formalize this question in the real world, and some additional ones when considering a simplified simulated setting. We discuss the most common and meaningful scenarios in Section 3.

Basic metrics are known in the machine learning community (and beyond) to pose problems; we investigate in Section 4, where we also address metrics based on information theory, such as mutual information.

Related work. Several security definitions have been proposed to quantify the level of anonymity provided by ACNs. While those definitions are suitable to argue about high latency, message-based ACNs (such as mix-nets and DC-nets), as we argue in Section 3.1, those definitions are less relevant to low latency, circuit-based onion routing like Tor.

Wagner and Eckhoff [57] provide an overview of possible metrics related to anonymity, including ACNs (see also Section 4). Some parallels exist between evaluating ACNs and side-channel attacks (SCA) [53], where a distinguisher wants to recover a subkey given a number of power traces: key recovery is essentially a classification problem and, as for traffic traces, the exact distribution of power traces is typically unknown, yet can be sampled from using real devices.

2 High-level Framework/Execution Environment

Setup. We apply a perspective of modern cryptology by describing the interaction between an adversary $\mathbb{A}$ and the ACN as a game, where our focus is on identifying the relevant random variables, as summarized in Table 1. For real-life ACNs, the distribution of these random variables might be unknown and difficult to estimate precisely; as we will see, an evaluator typically has far more control over the underlying distributions by using a semi-simulated experimental setting.

Central to our modelling are users who wish to connect to various destinations. We use disjoint sets $\mathcal{U}$ and $\mathcal{D}$ to describe, respectively, users and destinations. In the real-world, users and destinations are typically identified using IP addresses or URLs, possibly even names; abstractly any label suffices.

The choice of the users which destinations to connect to, is modelled by the random variable S, whose sample space is the set of directed bipartite graphs between $\mathcal{U}$ and $\mathcal{D}$. Nodes represent the users and destinations, whereas edges map connections between users and destinations.

The random variable S captures users' behaviour by selecting a single graph from the pool of possible ones, but this abstraction does have some shortcomings when compared to practice. Firstly, our model is static, in the sense that all users simultaneously decide on their destinations. In reality, users come and go and their destinations change over time. Such a dynamic setting seems to have received relatively little attention so far, hence our restriction to static appears standard. Secondly, an evaluator needs to choose a suitable distribution of S that is representative of real usage.

When evaluating, one often considers only simplified distributions for S. For instance, it is common that each user only connects to exactly one destination, so all user nodes have degree one. Additionally, either for all users the destination they connect to is independent and identically distributed (uniformly or based on website popularity metrics) or, in the special case where $|\mathcal{U}| = |\mathcal{D}|$, the graph S might correspond to a permutation, drawn uniformly at random.

For each user $u \in \mathcal{U}$, the destinations they are actually connecting to are denoted by random variable D_u. For some specific users there might be restrictions on the possible destinations, that is the support $\mathcal{D}_u$ of possible destinations is a proper subset of $\mathcal{D}$. Finally, destinations are said to be active if they have non-zero degree.

The state and all the possible observables of the anonymity network are modelled by G . In the case of Tor, G could capture the internal states of routers (identities, cryptographic keys, circuit IDs), traffic traces consisting of vectors of packet sizes and timings, and any other information that may be collected by any party involved (internal or external). Giving an exhaustive, formal definition of G is neither convenient nor necessary, though G should fully determine S.

Adversaries. Most adversaries will only have limited knowledge of G. Their view V of G depends on the specific threat model. For example, a user's ISP will be able to see that user's traffic patterns, but not much more. In contrast, the user's guard node will see that traffic pattern, but additionally know the middle router for that user's circuit. We will discuss threat models in more detail in Section 5.

An actual adversary often runs in two stages. During the first 'training' stage, it tries to learn general information about the behaviour of the network. For instance, how traffic traces captured at a proxy depend on the destination, or how traffic traces captured at a proxy differ from the corresponding ones captured at the exit node. This auxiliary information is captured by the random variable Z. Although Z could include an estimation of the distribution of S, it is independent of the random variable S itself. Only in the subsequent second 'challenge' stage the adversary observes V, which it combines with the auxiliary information Z to try and learn something useful about S.

What the adversary tries to learn corresponds to the goal of the adversary, which we capture by a query $\mathtt{q}$ that may depend on a target $\mathtt{T} \subseteq \mathcal{U} \times \mathcal{D}$. Given a query $\mathtt{q}$, the target $\mathtt{T}$ and the random variable S, there is often a unique answer to this query, which we will denote $S_{|\mathtt{q}}$ (with implicit dependency on $\mathtt{T}$). For instance, if $\mathtt{q}$ asks which website(s) a user u is connected to, then the target can be encoded as $\mathtt{T} = \{u\} \times \mathcal{D}$ and the correct, complete answer $S_{|\mathtt{q}}$ is a subset of $\mathcal{D}$. Note that unicity of the answer in the example above is a property of the query, irrespective of either target or instantiation of S.

When discussing possible goals in Section 3, we will refer to the users (resp. destinations) component of the target $\mathtt{T}$ as $\mathtt{T}_{\mathcal{U}}$ (resp. $\mathtt{T}_{\mathcal{D}}$), so in the small example above $\mathtt{T}_{\mathcal{U}} = \{u\}$ is relevant, whereas $\mathtt{T}_{\mathcal{D}} = \mathcal{D}$ is just a formalization artefact (and we may abuse the notation and consider either $\mathtt{T}_{\mathcal{U}}$ or $\mathtt{T}_{\mathcal{D}}$ to be empty instead).

The adversary processes the information and returns output $O_{\mathtt{q}}$ as response to the query $\mathtt{q}$ on target $\mathtt{T}$. If $\mathtt{q}$ has unique answers, this output $O_{\mathtt{q}}$ could be the adversary's best guess for $S_{|\mathtt{q}}$, or it could be an approximation, a list of possible answers, or a vector of likelihoods for select answers, etc.

Evaluation. When evaluating an attack, we specifically refer to the processing from V into O, possibly in conjunction with how Z is attained and used. On the other hand, when referring to a defence, we are primarily interested in how the random variable G's distribution can be made less susceptible to later attacks.

An evaluation should indicate how well an attack or defence works, which can be done by means of an anonymity metric. In first instance, such a metric is a parameter that summarizes the anonymity, or loss thereof, as indicated by the random variables S and O, or S and V (possibly also including Z). In that sense, a metric can be regarded as a population parameter (or as the difference between two population parameters, cf. the deltas used by Pfitzmann and Hansen [43]).

Although the distribution of G and thus V is typically unknown, one can sample from it, e.g. by connecting to the Internet using the ACN and taking measurements (in an ethically responsible way) or by using a tool like Shadow [31]. Thus, the population parameters can be estimated using sample statistics.

As we will see in Section 4, sometimes, metrics are simply expressed as summary statistics (e.g. accuracy), without reference to their potential underlying population parameter. We believe one strength of our framework of making the random variables explicit, is that it helps surface a number of otherwise hidden choices in the evaluation, such as the distribution of S used and how the sampling experiment was set up.

To evaluate an attack, i.e. the processing of V into $O_{\mathtt{q}}$, one inevitably has to take that output $O_{\mathtt{q}}$ into consideration. As the output is, or is related to, an adversary's best guess for $S_{|\mathtt{q}}$, such an output-dependent metric will depend on the adversary's goal. On the other hand, to evaluate a defence, changes in the distribution of V are more relevant, leading to metrics directly on V. Through $S_{|\mathtt{q}}$, such input-dependent metrics can still take into account an adversary's goal $\mathtt{q}$, yet without considering how the processing works. Thus, defences can potentially be evaluated independently of the currently best-known attacks.

Traffic traces as contained in V can contain a lot of unstructured data that is computationally expensive to process directly. An adversary may pre-process V by extracting its most salient features prior to the actual core processing. This core processing itself is often independent of any goal. For instance, for each observed ingress trace and each observed egress trace, it outputs a score how well they match, resulting in a matrix of scores. Subsequent post-processing can take this matrix to return an output specific to a given query `q`, say by turning the score into a true/false value (based on some threshold) or taking an arg max. Intuitively, such post-processing potentially throws away a lot of information, thus we might also want to consider metrics purely for the core-processing without taking into account any post-processing.

Examples. To illustrate our framework, see Example 1 (and Fig. 2) below.

Example 1 (Website Fingerprinting). Consider the *open world* scenario from Deep Fingerprinting by Sirinam *et al.* [52]. The threat model consists of a local passive adversary sniffing the traffic between a single user and their entry node. Such an adversary wants to verify whether the user is accessing a website from some pre-defined subset $\mathtt{T}_{\mathcal{D}} \subsetneq \mathcal{D}$ or not. In our framework, this translates to:

- S The website w accessed by the user, sampled from $\mathcal{D}$ according to a uniform distribution. Using the graph notation, $\mathcal{U} = \{u\}$ and there is a single edge going from u to w.
- G The state of the single circuit in the network: identities of the user, nodes, and destination, plus traffic flows from user to destination and vice versa.
- V The information in G accessible by the adversary, mainly user's identity and the traffic trace between user and guard node.
- **goal** The query "Is the visited website w an item of $\mathtt{T}_{\mathcal{D}}$?", where $\mathtt{T}_{\mathcal{D}}$ is the target.
- Z The training data used by the adversary's distinguisher. It is referred to as *open-world dataset*.
- O A binary random variable, answering directly the query `q` with `yes` or `no`.

This process is performed on a single traffic trace at a time, resulting in the binary random variable O. The experiment is repeated in order to collect more data and extrapolate statistical information. *Precision* and *recall* are used to estimate the performance of the distinguisher; moreover, ROC curves are provided to show the trade-off between TPR and FPR.

Example 2 (in the appendix) looks at the traffic analysis scenario studied by Nasr *et al.* [37]. There is a key difference between those two examples. In the first one, the adversary attacks a single user at a time, in a way that allows them to scale their approach to multiple users independently of each other. Consequently, the influence on how well the attack performs against a specific user is largely independent of what other users are doing, seemingly contradicting the intuition that the anonymity in onion routing derives from 'hiding among the masses' [19]. In contrast, for the second example all users' traffic traces are pooled together and users may influence each other's anonymity.

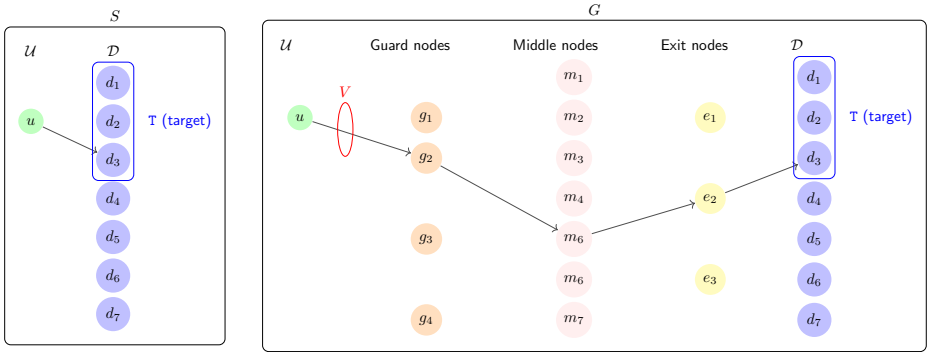


Fig. 2. Bipartite and multipartite graphs representing settings and random variables in Deep Fingerprinting (Example 1). Edges in the secret S are expanded, through protocol specifications, to multiple edges in the multipartite graph G (representing circuits in Tor) and the view of the adversary V.

Relevance. When we say that we focus on evaluation attacks and defences against the anonymity of Tor, our scope is relatively narrow, as we primarily concentrate on fingerprinting and traffic analysis in a scenario where the threat model is fixed (see also Section 5). However, an adversary who controls all routers in the Tor network can trivially de-anonymize all circuits, whereas an adversary who controls no routers, proxies, or destinations and cannot observe any traffic, essentially for whom $V = \emptyset$, cannot possibly learn anything useful.

A rational de-anonymization adversary might therefore invest its resources in observing and controlling a chunk of the ACN as large as possible. For a user worried about deanonymisation through ingress–egress traffic analysis, the largest risk arguably lies in the adversary capturing both traffic traces, and less in the adversary's ability to link the traces (if both ends had been acquired within a much larger collection of traces). On the opposite side, deployers of an ACN would probably spend more of their resources in protecting their network from control and widespread observation, rather than incorporating bandwidth-consuming and latency-increasing countermeasures to reduce the damage if an adversary sees both ingress and egress traces [55].

We acknowledge that, rather than minimizing the damage when a threat occurs, it makes sense to minimize the threat happening in the first place and the Tor designer's assumption that all anonymity is lost if both guard and exit node of a circuit are compromised is the correct conservative one. Similarly, guard nodes were introduced in recognition that it is better to sacrifice a few users more severely, than a lot of users a little. Yet that does not take away that figuring out how to compromise in various scenarios, and how to limit those compromises potentially, is an active research area. Whether to deploy a potential defence in practice will be based on its return on investment, essentially a trade-off between the protection offered versus the cost (in comparison with other measures outside the scope of this work).

3 Security Goals and Notions

3.1 Interpreting Privacy Notions

Historical context. After Chaum [11] initiated the study of ACNs in the early 1980s, it quickly became apparent that a common language was lacking. Pfitzmann and Hansen [43, 44] attempted to consolidate terminology by providing as precise as possible context-free descriptions of various relevant terms, such as unlinkability and unobservability. For specific contexts, they recommend to abstract away certain terms, such as 'sender', 'recipient', and 'message'.

In the context of onion routing, the user establishes a circuit over which bidirectional traffic will flow, making the concepts of sender, receiver, and especially message potentially misleading. Using the existing, default terminology as is might encourage a mental model of a sender sending a single or vector of messages to a receiver. Such a mental model could lead to mismatches in context-specific formalizations, similar to a message-based mental model not quite capturing TLS's record layer security [24]. Thus we speak of users instead of senders and destinations instead of receivers; furthermore, we drop the concept of messages from our framework, arguably the closest analogy would be a circuit.

The privacy notions of unlinkability and unobservability have been formalized for ACNs in a number of works [2, 7, 27]. In particular, Kuhn *et al.* [34] present a thorough formalisation of a wide range of privacy notions, encompassing most previous work. For high latency message-based ACNs, these indistinguishability-based notions are very suitable as they allow expressing (and proving) the security of protocols in a fine-grained manner. However, the notions show some shortcomings when it comes to their applicability to low latency ACNs, such as the inherent dichotomy between success or failure of the attacker in terms of *whether* a formal definition is satisfied or not. This approach is rarely encountered in the literature concerning onion routing in the real world, which commonly rely on measuring *how well* attacks and countermeasures perform. Moreover, they consider asymptotic security as they employ adversaries as probabilistic polynomial time (PPT) algorithms, instead of concrete real-world instantiations.

Our interpretation. Minding the above, we depart from the indistinguishability-based formalizations and provide an interpretation, specific for onion routing, of observability and linkability (O, L), usually in their negated forms unobservability ($\overline{\mathsf{O}}$) and unlinkability ($\overline{\mathsf{L}}$). Instead of sender and receiver, we maintain our terminology of users (U) and destinations (D). Note that we abstract away any particular onion routing specifications, so our notions are agnostic of, for instance, the use of guard nodes, or the length of the circuits.

Tor's ultimate goal is to avoid any party, different from the user themselves, from learning both user and destination of observed traffic. The corresponding privacy notion is then user–destination unlinkability ($(\mathsf{UD})\overline{\mathsf{L}}$). User unobservability ($\mathsf{U}\overline{\mathsf{O}}$) refers to the inability, for the adversary, to observe whether a user is accessing Tor or not. This notion is important, for example, in cases where

Table 2. Privacy notions. A node v of the graph S is active if and only if $\deg(v) > 0$.

Notion	Description
$(\mathsf{UD})\overline{\mathsf{O}}$	**1.** No edges can be noticed from either users or destinations; the number of edges can be disclosed. This condition can be expressed as unknown degree value for any of the nodes but known total degree of the graph ($\lvert E_S \rvert$).
$(\mathsf{UD})\overline{\mathsf{L}}$	**2.** Degrees of nodes are revealed, but no element e of the edge set E_S is known. In terms of S, no path is completely disclosed, but which users or destinations are active can be revealed.
$\mathsf{U}\overline{\mathsf{O}}$	**3.** Degree of destinations is known, but not for users'.
$\mathsf{D}\overline{\mathsf{O}}$	**4.** Which destinations are active is unknown; instead, users' activity may be disclosed.

Tor usage is being censored [20, 21, 60–62]; the analogous notion ($\mathsf{D}\overline{\mathsf{O}}$) can be considered for destinations. These two notions could also be combined into user–destination unobservability ($(\mathsf{UD})\overline{\mathsf{O}}$), in case neither the user nor the destination can be observed as being connected to the Tor network.

Above, when referring to privacy notions, we only provided intuitive descriptions rather than the formal definition approach mentioned previously. In Table 2 we interpret the privacy notions in terms of the bipartite graph representing S. For example, observing a destination means knowing that the destination is connected to the network, i.e. the corresponding node in the graph has non-zero degree. We will refine further when discussing privacy goals in the next section.

Relationships. Ostensibly, unobservability is a stronger notion than unlinkability (cf. [34, 43]). Yet, somewhat counterintuitively when we consider specific threat models against onion routing, it appears that the seemingly stronger looking abstraction (i.e. unobservability) can be the more appropriate. Let us elaborate.

Depending on the threat model (Section 5), $(\mathsf{UD})\overline{\mathsf{L}}$ may collapse to either $\mathsf{U}\overline{\mathsf{O}}$ or $\mathsf{D}\overline{\mathsf{O}}$. For example, assume that the adversary observes the traffic between the user and the guard node, either by corrupting the guard node or by observing traffic in the user's or the guard node's ISP. These observations reveal the user's IP address as well as the traffic patterns from and to the user. Based on this information alone, website fingerprinting may in some cases (with a well fingerprinted server) help the adversary to identify the server and hence the user and destination are linkable. However, if in addition the adversary is able to observe the traffic between the exit node and the destination, it is highly plausible that linking the user and the destination is computationally feasible. Hence, in this scenario, in order to achieve $(\mathsf{UD})\overline{\mathsf{L}}$, we need the adversary to be unable to observe the destination of the traffic ($\mathsf{D}\overline{\mathsf{O}}$). Note that, as it is customary in the literature, such terminology ($\overline{\mathsf{O}}$) does not take into account the computational effort needed to infer the desired information (e.g. the end destination) from the

available information (e.g. the traffic trace), instead of distinguishing between the concepts of 'unobservability' and 'computational uninferrability'.

3.2 A Taxonomy of Security Goals

Attacks on user–destination anonymity can have different goals, as captured in our framework in terms of queries `q` on S, where `q` may furthermore have a specific target `T`. Distinct goals may require different metrics, which are reflected in the literature, where authors utilise various evaluation metrics to compare their results with others. In order to understand the various metrics, we first need to establish a taxonomy of different goals, which we will do in this section. We divide the goals and the corresponding queries in four distinct categories. From specific to the most general these are distinguishing, decisional, classification, and finally computational. Examples 3–6 in Appendix B illustrate these goals.

Distinguishing goals. Inspired by the classic IND-CPA [6] notion for encryption, distinguishing goals arise in formal cryptologic models of anonymity (see also Section 3.1): an adversary is interacting with one of two worlds (say left or right) and needs to figure out which world it is engaged with. A distinguishing goal corresponds to a dichotomous classification problem with a uniform prior and symmetry between the two classification options, with no meaningful distinction between positives and negatives (see Example 3). Assuming the output is a single bit, the typical metric for a distinguishing goal is the distinguishing advantage.

Decisional goals. Decisional goals are still dichotomous classification problems, but here the prior might be non-uniform and meaning can be associated to positives and negatives. The open world scenario (Example 1) belongs to this category, since $\mathcal{D}$ can be partitioned in monitored/unmonitored destinations and the adversary has to decide whether the observed traffic trace corresponds to a monitored destination or not, without having to pinpoint the exact destination.

In our framework, a query `q` representing a decisional goal can be regarded as a predicate on S that induces a partition of S's sample space into two subsets: the positive part contains all bipartite graphs satisfying the predicate (e.g., the targeted user connects to a monitored website), whereas for negatives the predicate is false (the user is in the clear). As the concepts of true/false positives/negatives are meaningful, the standard metrics for binary classifiers apply, which we will expand upon in Section 4.2.

Decisional goals are often related to the open world scenarios in website fingerprinting, introduced by Panchenko *et al.* [42]. They suggest it as it is closer to a real world scenario (compared to closed world), and it quickly became one of the two main instantiations of website fingerprinting [8,10,26,30,41,48,52,58,59].

Classification goals. For a more general classification goal, we drop the requirement of only two classes available, thus the query `q` induces a partition of

the sample space of S in more than two subsets. A key difference compared to decisional goals is that, from the adversary's perspective, there is no longer any preference among the possible classes and specifically all misclassifications are treated the same (in sharp contrast to false positives versus false negatives for decisional goals). In that sense, classification goals are closer to distinguishing goals, however for classification goals the prior distribution need not be uniform over the classes.

The closed-world scenario for website fingerprinting is an example of a classification goal; another example arises in traffic analysis attacks when an adversary has to match a single target ingress trace to one of many possible egress traces, or vice versa [54]. Accuracy, corresponding to the probability of classifying correctly, is the most common metric.

Computational goals. Finally, for computational goals an adversary tries to learn something that perhaps cannot easily be classified and there might not be a single correct answer. For instance, a greedy adversary trying to deanonymize as many users simultaneously as possible. Often there is a notion of proximity or similarity between answers, including not-quite-correct ones, rendering the adversary's job one of best-effort estimation.

A typical example of a computational goal is the matching of ingress traces to egress traces [25,36,37,40,50,54]. Although there is a unique best bipartite graph that correctly identifies all matches without any incorrect ones, an adversary might prefer to only output the matches it is most confident in, or it might even output an inconsistent set of matches in order not to miss any legitimate matches. See also Example 6.

Discussion. Some goals that are seemingly identical can be modelled in slightly different ways when evaluating. For instance, when an evaluator is interested how well a website fingerprinting algorithm works on ingress traces, one option is to consider multiple users and target only one (so V is considerable smaller than G), another is to only ever consider a single user (so V contains more of G). When the traces are acquired by live interaction with the ACN (including many unknown users outside $\mathcal{U}$), the two views V might be sufficiently similar to render the single-user simplification representative of the multi-user setting. If, on the other hand, traces are simulated, simplifying away other users may not be warranted.

Goals can also relate to each other in a black-box way, in the sense that an adversary that decides whether an ingress trace and an egress trace are related, might also be used to determine which ingress trace belongs to a given egress trace by selecting one of the matching ingress traces (ideally, there would be exactly one, but this cannot be guaranteed). However, such a black-box approach is likely wasteful if the adversary really creates a score as its core processing and only arrived at a yes/no decision through post-processing. In that case applying a different post-processing instead makes more sense.

We list a number of possible goals in Table 4, Appendix C.

4 Metrics

Syverson [55] argues that anonymity metrics should reflect the effort an adversary has to expend in order to reach a goal, and also that to be useful, security metrics should not depend on the values of variables for which we cannot make adequate relevant determinations or predictions. We believe anonymity metrics should be suitable for the security goal at hand, they should allow meaningful comparison between different attacks and countermeasures, and efficient and robust estimation should be feasible. In order to be sufficiently general to accommodate a wide range of goals and attacks, we use the term "metric" in a relaxed manner, without imposing the usual mathematical properties of a metric. What we will assume is that a metric μ does not behave in a non-intuitive way.

Researchers on ACNs like Tor often have limited access to real world data, due to intrinsic difficulties including legal and ethical considerations. Hence, assumptions known to be artificial are regularly employed but seldom explicitly stated. Furthermore, since metrics are used to represent the performance of an attack, they depend on the input data collected by the adversary or by the evaluator for the given attack and not only on the attack itself. The accuracy of a classifier may be influenced by the probability distribution on the input data, by the size of the data set, or by the number of classes. In consequence, metrics used in the literature are often effectively estimates of the adversary's success rate in synthetic settings, while their real world relevance remains less clear [56].

For an attacker, the relevant metrics may be the computational cost and accuracy of an attack, while researchers may be interested in the anonymity level provided by Tor to the average user. Due to these substantial differences, it is of fundamental importance to determine which metrics are pertinent rather than defaulting to some generic ones.

We provide examples illuminating the concepts of this section in Appendix D.

4.1 Input-dependent Metrics

Input-dependent metrics depend on S or $S_{|\mathsf{q}}$, G, and V, but not on O. The leakage about the secret random variables S, $S_{|\mathsf{q}}$, and G obtained by observing V is naturally expressed in the form of information-theoretic concepts like entropy, conditional entropy, and mutual information [35]. These concepts have already been used to assess anonymity networks [4, 16, 49, 51]. The Shannon entropy

$$\mathrm{H}(S) = -\sum_{s \in S} p_S(s) \log_2 p(s)$$

by itself only expresses the a priori uncertainty about the secret S. In order to evaluate an attack, it is necessary to study the conditional entropy

$$\mathrm{H}(S \mid V) = -\sum_{s \in S, v \in V} p(s, v) \log_2 p(s|v) = \mathbb{E}[\mathrm{H}(S \mid V = v)]_{v \in V}, \qquad (1)$$

which represents the remaining uncertainty about S after observing the random variable V. Hence, the mutual information $\mathrm{I}(S; V) = \mathrm{H}(S) - \mathrm{H}(S \mid V)$ can

be interpreted as the information leakage about S from observing V. Shannon entropy is known to satisfy *monotonicity*, that is, $\mathrm{H}(S \mid V) \leq \mathrm{H}(S)$, so that information leakage defined as above is always non-negative.

Diaz *et al.* [17] remark that since the RHS of eq. (1) contains an expectation, there may exist some sample view v that gives more leakage than average, and hence one might be concerned about sample views of this type.

In our model, the view V is a random variable beyond the influence of the adversary, and in a pure Shannon entropic perspective, only the average conditional entropy $\mathrm{H}(S \mid V)$ would be important. However, concern about deanonymisation probabilities suggests that more emphasis should be put on this type of sample views (cf. [43, Footnote 34]). Hence, Clauß and Schiffner [13] suggested the use of Rényi entropy and quantiles to measure anonymity. Rényi entropy [1,23,47] is a generalization of Shannon entropy (for convenience, we include a brief summary of relevant concepts in Appendix D), which Clauß and Schiffner argue is more resilient (using different values of α) against the influence of outliers than Shannon entropy. They differentiate between *network* and *application* layers when assessing anonymity, corresponding respectively to G and S in our framework.

In order to compare the performance of attacks in different settings with different sizes of the secret S, some authors [16,57] suggest *normalizing* information metrics by dividing by the secret max entropy $\mathrm{H}_0(S) = \log_2|S|$.

4.2 Output-dependent Metrics

By definition, the adversary view V contains all of the information the adversary obtains about the secret S, or about the parts $S_{|\mathsf{q}}$ of the secret pertinent to a specific query q. However, V is typically complex and the appropriate response to q may not be immediately obvious based on inspection of V. In order to provide an illuminating response O_{q} that is aligned with the query q, the adversary needs to apply a query-dependent processing of V. An evaluator with access to the secret S (or $S_{|\mathsf{q}}$) should be able to compute an *output-dependent* metric function $\mu_{\mathsf{q}}(O_{\mathsf{q}}, S)$ that measures the quality of the (estimated) output O_{q} relative to S.

It follows by the data processing lemma of information theory [14, Section 2.8] that $\mathrm{I}\left(S; O_{\mathsf{q}}\right) \leq \mathrm{I}(S; V)$ (and similar for Rényi information). Moreover, since V is typically complex and machine learning may be part of the processing, outputs may be unaccompanied by confidence/uncertainty estimates, and thus, the price for the adversary of providing an output O_{q} in a convenient form is often an information loss. The examples in Appendix D illustrate this.

In the literature, authors have used various output-dependent metrics to quantify the success of attacks. A general consensus on which of these are more insightful for a given type of query still appears lacking. Table 5 briefly describes some metrics that have been used in the context of onion routing attacks.

Decisional goals. Queries leading to decisional goals can be considered analogous to binary classifiers, for which the concepts of true/false positive/negative are clear. Two common metrics used for such queries are *Precision* and *Recall*,

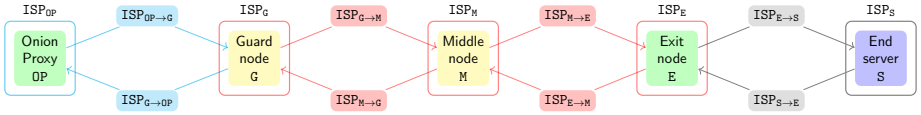


Fig. 3. Example of Tor circuit. As discussed by Sun *et al.* [54], the traffic path may be asymmetric in the forward and backward directions.

defined in Table 5. ROC (Receiver operating characteristic) curves [22] provide a more comprehensive description of performance.

Caveat: A binary classifier with non-zero false positive rate which is applied to a random variable with a very low prior probability of being positive suffers from the so called *Base Rate Fallacy*, by which most positive outputs will be incorrect. This scenario has been applied to Tor [32, 56], highlighting potential disadvantages of these metrics and calling for more precise description of the setup of experiments and presentation of the results.

Classification goals. Metrics for classification goals cannot rely on the difference between positive and negative guesses, but only on guesses being either correct or wrong. Thus, they are susceptible to biases in the data sets: for example, if the prior distribution of the data set on three possible classes is $\{0.9, 0.05, 0.05\}$, a naïve classifier with constant output '`class 1`' has 90% accuracy.

Computational goals. Computational goals represent the most general case in our framework and are characterised by the concept of "closeness" of the output to the real answer to the query, i.e. guesses by the adversary may be partially correct—in a similar way to fuzzy logic truth values.

5 Adversarial Threat Model

Threat modelling is a central part of the analysis of security and anonymity; we consider the following general adversary characterisations [57] for onion routing:

- *passive* adversaries are only allowed to observe the protocol execution, and as such they can be thought as *honest-but-curious*. *Active* adversaries, on the other hand, can modify, delay, replay, stop the traffic. *Semi-honest* adversaries are a relevant subset of the latter category: they tamper with the traffic in a non-disruptive way only, e.g. by slightly delaying the cells;
- only *internal* adversaries have access to data inside onion nodes, while *external* ones are limited to non-onion nodes. We will refer to adversaries having access to both onion nodes and external parties as *hybrid*;
- *local* adversaries control (observe) only some of the nodes of the network, while *global* ones do not have this limitation. For example, a global internal adversary controls all and only the Tor nodes and global external has access to all and only the Internet infrastructure.

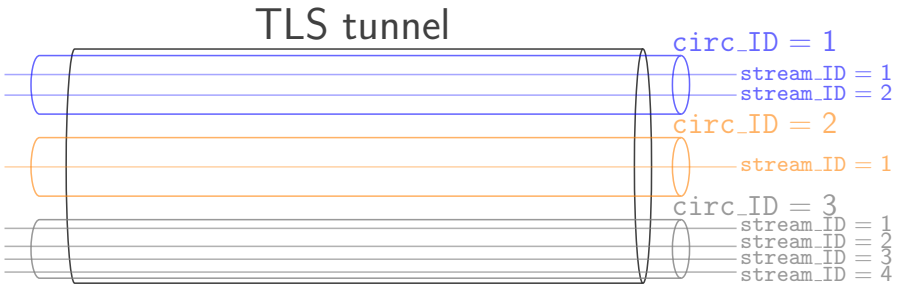


Fig. 4. Multiplexing in onion routing.

Each of these characteristics is orthogonal to the others, and the terminology reflects these degrees of freedom: passive adversaries tend to *observe* nodes, while active ones *control* them.

According to this characterisation, an active global hybrid adversary is the most powerful. Goldschlag *et al.* [29] explicitly state that onion routing does not aim to protect from global adversaries, regardless of the computational cost of processing all the information or if they are internal or external. On the other hand, local adversaries have a restricted view of the network and they should not be able to link user and destination of the traffic.

While a local adversary is the standard in the literature, other characteristics vary among the different works, with external and passive adversaries being the most common [37, 38, 40, 41, 48, 52, 58]. Some results require the adversary to be not only internal but also semi-honest [36, 50, 54] or even active [25].

Taking into consideration the Tor protocol specifications [18], it is worth noting that even the same type of adversary may have a more or less granular view of the traffic, depending on their position along the circuit. This is due to the fact that Tor multiplexes traffic on two layers: a single onion circuit can carry several streams (i.e. TCP connections) and nodes multiplex several circuits on the same TLS tunnel (Fig. 4).

In general, internal adversaries have access to more information compared to externals (e.g. single cells, `circ_ID`), but they can be detected and removed from the network [3]. External adversaries observe only TLS-encrypted tunnels but Tor tends, to the proxy, to stream single cells in TLS packets, so they can be inferred by some ISP on the circuit (e.g. $\texttt{ISP}_{\texttt{OP}}$ in Fig. 3).

Assuming for simplicity that the onion proxy creates a single circuit, this is the path from the user to the end destination (cf. Fig. 3):

1. $\texttt{ISP}_{\texttt{OP}\to\texttt{G}}$ and $\texttt{ISP}_{\texttt{G}\to\texttt{OP}}$ observe a single TLS tunnel carrying the user's circuit;
2. $\texttt{ISP}_{\texttt{G}}$ (and all the others in red in the figure) observe TLS tunnels, possibly carrying multiple circuits;
3. the guard node `G` and the middle node `M` observe single cells, but not streams;

4. $\texttt{ISP}_\texttt{E}$ is different, in that they observe also the non-Tor traffic directed to the end destinations. In case of browsing, though, this traffic tends to be encrypted as well [39];
5. exit node `E` is the only one to have visibility on the `stream_ID` as well, distinguishing different TCP connections originating from `OP`.

Depending on the settings, this design will prevent many types of adversary from reaching their goal. For example, assume that two users simultaneously create a single circuit each, passing through the same nodes, to different destinations: then, no external adversary can distinguish the streams between the exit node and the end destinations.

6 Application of Our Framework

Our framework allows to clearly describe assumptions underlying attacks and evaluation of anonymity in onion routing; we present a brief list of existing literature expressed in terms of our framework in Table 4.

First, S, G and V are defined taking into consideration the assumptions about the environment. Both the supports of those random variables and their probability distribution are needed to completely characterise the experiment. Furthermore, the definition of V guarantees that also the adversarial model is well specified and understood. The last step of the setup phase is the determination of the query `q` and optionally a target `T`. These, on the other hand, influence which type of random variable $O_\texttt{q}$ to employ.

Such process ensures all the game variables are well defined, allowing to establish which type of goal the adversary is trying to achieve. Finally, each case may require different metrics to meaningfully and effectively illustrate the results, while making sure that limitations are apparent and common misunderstanding avoided.

7 Conclusion

We highlighted several of the challenges when evaluating onion routing and described a framework that helps to benchmark different attacks and countermeasures. Although we did not explicitly mention all features of Tor, we expect that for instance so-called leaky pipes and hidden services can be easily integrated into our framework.

We leave open the dynamic situation, where users come and go, and an adversary might actively try to influence the (re)establishment of circuits. Formally making the various random variables time-dependent is easy enough, simply by writing $S(t)$ instead of S. However, one main challenge we see are determining meaningful, possibly adversarially affected, evolutions of the secret $S(t)$. A fixed uniform distribution as often used for a static S somewhat defeats the purpose of the dynamic setting, but could still serve a situation where an adversary can trigger (as in Tor) circuit teardowns that are subsequently re-established, making the overall view of the system G depend on the adversary.

Table 3. Attacks and their instantiation in our framework. Attacks' names have been assigned by the authors of this work for convenience. Adversaries are assumed to be local, for the other features the corresponding initial letter is used. We refer to Table 4 for goals.

Ref.	Attack	Adv.	Goal type	Output	Metric
[40]	Wavelet Multi-resolution	E, P	Computational (7)	Pairs (ingress, egress) flows	FP/FN
[50]	Dropmarking	I, S-H	Classification (5)	Pairs (entry, exit) nodes	TPR/TNR/FPR/FNR
[37]	DeepCorr	E, P	Computational (7)	Pairs (ingress, egress) flows	FPR/FNR, ROC curve
[54]	RAPTOR	E, P	Classification (5)	Pairs (ingress, egress) flows	Accuracy
[36]	Cell counting	I, S-H	Classification (4, 5)	Pairs (ingress, egress) flows	Accuracy, detection rate, false positive rate
[25]	Tagging attack	I, A	Decisional (3)	Pairs (ingress, egress) flows	Accuracy
[45]	Fingerprinting with website oracles	E, P	Decisional (3), Classification (4)	Pairs (trace, website)	Precision, recall
[52]	Deep Fingerprinting	E, P	Decisional (3), Classification (1)	Pairs (trace, website)	Accuracy
[48]	Fingerprinting with Deep Learning	E, P	Decisional (3), Classification (1)	Pairs (trace, website)	Accuracy, TPR/FRP,ROC curve
[30]	Correlation with DNS info	E, P	Computational (7)	Intersection of ASes' sets	Precision, recall
[38]	Compressive Traffic Analysis	E, P	Computational (7)	Pairs (flow, noisy flow), pairs (trace, website)	TP/FP, accuracy
[41]	Fingerprinting at Internet scale	E, P	Decisional (3), Classification (1)	Pairs (trace, website)	Accuracy, precision
[58]	k-NN Website Fingerprinting	E, P	Decisional (3), Classification (4)	Pairs (trace, website)	TPR/FPR, accuracy
[10]	Circuit clogging	I, S-H	Classification (1)	Pairs (user, onion nodes)	TPR/FRP, ROC curve
[59]	SVM Fingerprinting	E, P	Decisional (3), Classification (1)	Pairs (trace, website)	Accuracy, TP/FP
[26]	Induced throttling	I, S-H	Classification (1)	Pairs (user, onion nodes)	Percentile, degrees of anonymity, client probability
[8]	DLSVM	E, P	Decisional (3)	Pairs (trace, website)	TPR/FPR, success rate
[42]	Website fingerprinting	E, P	Decisional (3), Classification (1)	Pairs (trace, website)	Accuracy, TPR/FPR

References

1. Arimoto, S.: Information measures and capacity of order α for discrete memoryless channels. In: Topics in Information Theory. Colloquia Mathematica Societatis János Bolyai, vol. 16, pp. 41–52 (1977)
2. Backes, M., Kate, A., Manoharan, P., Meiser, S., Mohammadi, E.: AnoA: A framework for analyzing anonymous communication protocols. In: Cortier, V., Datta, A. (eds.) CSF 2013 Computer Security Foundations Symposium. pp. 163–178. IEEE Computer Society Press (2013). https://doi.org/10.1109/CSF.2013.18
3. Bagueros, I.: Tor security advisory: exit relays running sslstrip in may and june 2020 (Aug 2020), https://blog.torproject.org/bad-exit-relays-may-june-2020
4. Barton, A., Wright, M., Ming, J., Imani, M.: Towards predicting efficient and anonymous tor circuits. In: Enck, W., Felt, A.P. (eds.) USENIX Security 2018. pp. 429–444. USENIX Association (Aug 2018)
5. Bauer, K.S., McCoy, D., Grunwald, D., Kohno, T., Sicker, D.C.: Low-resource routing attacks against Tor. In: Ning, P., Yu, T. (eds.) WPES 2007. pp. 11–20. ACM, New York, NY, USA (Oct 2007). https://doi.org/10.1145/1314333.1314336
6. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: 38th FOCS. pp. 394–403. IEEE Computer Society Press (Oct 1997). https://doi.org/10.1109/SFCS.1997.646128
7. Bohli, J.M., Pashalidis, A.: Relations among privacy notions. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 362–380. Springer, Heidelberg (Feb 2009)
8. Cai, X., Zhang, X.C., Joshi, B., Johnson, R.: Touching from a distance: website fingerprinting attacks and defenses. In: Yu, T., Danezis, G., Gligor, V.D. (eds.) ACM CCS 2012. pp. 605–616. ACM Press (Oct 2012). https://doi.org/10.1145/2382196.2382260
9. Camenisch, J., Lysyanskaya, A.: A formal treatment of onion routing. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 169–187. Springer, Heidelberg (Aug 2005). https://doi.org/10.1007/11535218_11
10. Chan-Tin, E., Shin, J., Yu, J.: Revisiting circuit clogging attacks on Tor. In: ARES 2013. pp. 131–140. IEEE Computer Society (Sep 2013). https://doi.org/10.1109/ARES.2013.17
11. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the Association for Computing Machinery **24**(2), 84–90 (Feb 1981). https://doi.org/10.1145/358549.358563
12. Chaum, D.: The dining cryptographers problem: Unconditional sender and recipient untraceability. Journal of Cryptology **1**(1), 65–75 (Jan 1988). https://doi.org/10.1007/BF00206326
13. Clauß, S., Schiffner, S.: Structuring anonymity metrics. In: Juels, A., Winslett, M., Goto, A. (eds.) WDIM 2006. pp. 55–62. ACM (Nov 2006). https://doi.org/10.1145/1179529.1179539
14. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, Ltd, USA (2006). https://doi.org/10.1002/047174882X
15. Das, D., Meiser, S., Mohammadi, E., Kate, A.: Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency - choose two. In: 2018 IEEE Symposium on Security and Privacy. pp. 108–126. IEEE Computer Society Press (May 2018). https://doi.org/10.1109/SP.2018.00011

16. Díaz, C., Seys, S., Claessens, J., Preneel, B.: Towards measuring anonymity. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 54–68. Springer, Heidelberg (Apr 2002). https://doi.org/10.1007/3-540-36467-6_5
17. Díaz, C., Troncoso, C., Danezis, G.: Does additional information always reduce anonymity? In: Ning, P., Yu, T. (eds.) WPES 2007. pp. 72–75. ACM, New York, NY, USA (Oct 2007). https://doi.org/10.1145/1314333.1314347
18. Dingledine, R., Mathewson, N.: Tor protocol specification (Aug 2021), commit 6d1e05d, https://raw.githubusercontent.com/torproject/torspec/c17c36c57635a9ebf88b2b41dc41cbddcf56f7ef/tor-spec.txt
19. Dingledine, R., Mathewson, N., Syverson, P.F.: Tor: The second-generation onion router. In: Blaze, M. (ed.) USENIX Security 2004. pp. 303–320. USENIX Association (Aug 2004)
20. Dyer, K.P., Coull, S.E., Ristenpart, T., Shrimpton, T.: Protocol misidentification made easy with format-transforming encryption. In: Sadeghi, A.R., Gligor, V.D., Yung, M. (eds.) ACM CCS 2013. pp. 61–72. ACM Press (Nov 2013). https://doi.org/10.1145/2508859.2516657
21. Ensafi, R., Winter, P., Mueen, A., Crandall, J.R.: Analyzing the great firewall of china over space and time. PoPETs **2015**(1), 61–76 (Jan 2015). https://doi.org/10.1515/popets-2015-0005
22. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters **27**(8), 861–874 (2006). https://doi.org/https://doi.org/10.1016/j.patrec.2005.10.010
23. Fehr, S., Berens, S.: On the conditional Rényi entropy. IEEE Transactions on Information Theory **60**(11), 6801–6810 (2014). https://doi.org/10.1109/TIT.2014.2357799
24. Fischlin, M., Günther, F., Marson, G.A., Paterson, K.G.: Data is a stream: Security of stream-based channels. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 545–564. Springer, Heidelberg (Aug 2015). https://doi.org/10.1007/978-3-662-48000-7_27
25. Fu, X., Ling, Z.: One cell is enough to break Tor's anonymity (2009)
26. Geddes, J., Jansen, R., Hopper, N.: How low can you go: Balancing performance with anonymity in tor. In: De Cristofaro, E., Wright, M.K. (eds.) PETS 2013. LNCS, vol. 7981, pp. 164–184. Springer, Heidelberg (Jul 2013). https://doi.org/10.1007/978-3-642-39077-7_9
27. Gelernter, N., Herzberg, A.: On the limits of provable anonymity. Cryptology ePrint Archive, Report 2013/531 (2013), https://eprint.iacr.org/2013/531
28. Goldschlag, D.M., Reed, M.G., Syverson, P.F.: Hiding routing information. In: Anderson, R.J. (ed.) IWIH 1996. LNCS, vol. 1174, pp. 137–150. Springer, Heidelberg, Berlin, Heidelberg (Jun 1996). https://doi.org/10.1007/3-540-61996-8_37
29. Goldschlag, D.M., Reed, M.G., Syverson, P.F.: Onion routing. Communications of the Association for Computing Machinery **42**(2), 39–41 (Feb 1999). https://doi.org/10.1145/293411.293443
30. Greschbach, B., Pulls, T., Roberts, L.M., Winter, P., Feamster, N.: The effect of DNS on tor's anonymity. In: NDSS 2017. The Internet Society (Feb / Mar 2017)
31. Jansen, R., Hopper, N.: Shadow: Running Tor in a box for accurate and efficient experimentation. In: NDSS 2012. The Internet Society (Feb 2012)
32. Juárez, M., Afroz, S., Acar, G., Díaz, C., Greenstadt, R.: A critical evaluation of website fingerprinting attacks. In: Ahn, G.J., Yung, M., Li, N. (eds.) ACM CCS 2014. pp. 263–274. ACM Press (Nov 2014). https://doi.org/10.1145/2660267.2660368

33. Karunanayake, I., Ahmed, N., Malaney, R., Islam, R., Jha, S.: Anonymity with Tor: A survey on Tor attacks (2020), https://arxiv.org/abs/2009.13018
34. Kuhn, C., Beck, M., Schiffner, S., Jorswieck, E.A., Strufe, T.: On privacy notions in anonymous communication. PoPETs **2019**(2), 105–125 (Apr 2019). https://doi.org/10.2478/popets-2019-0022
35. Li, S., Guo, H., Hopper, N.: Measuring information leakage in website fingerprinting attacks and defenses. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 1977–1992. ACM Press (Oct 2018). https://doi.org/10.1145/3243734.3243832
36. Ling, Z., Luo, J., Yu, W., Fu, X., Xuan, D., Jia, W.: A new cell-counting-based attack against Tor. IEEE/ACM Transactions on Networking **20**(4), 1245–1261 (Aug 2012). https://doi.org/10.1109/TNET.2011.2178036
37. Nasr, M., Bahramali, A., Houmansadr, A.: DeepCorr: Strong flow correlation attacks on tor using deep learning. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 1962–1976. ACM Press (Oct 2018). https://doi.org/10.1145/3243734.3243824
38. Nasr, M., Houmansadr, A., Mazumdar, A.: Compressive traffic analysis: A new paradigm for scalable traffic analysis. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 2053–2069. ACM Press (Oct / Nov 2017). https://doi.org/10.1145/3133956.3134074
39. Naylor, D., Finamore, A., Leontiadis, I., Grunenberger, Y., Mellia, M., Munafò, M., Papagiannaki, K., Steenkiste, P.: The Cost of the "S" in HTTPS. In: Seneviratne, A., Diot, C., Kurose, J., Chaintreau, A., Rizzo, L. (eds.) CoNEXT 2014. pp. 133–140. CoNEXT '14, ACM (Dec 2014). https://doi.org/10.1145/2674005.2674991
40. Palmieri, F.: A distributed flow correlation attack to anonymizing overlay networks based on wavelet multi-resolution analysis. IEEE Transactions on Dependable and Secure Computing (To appear). https://doi.org/10.1109/TDSC.2019.2947666
41. Panchenko, A., Lanze, F., Pennekamp, J., Engel, T., Zinnen, A., Henze, M., Wehrle, K.: Website fingerprinting at internet scale. In: NDSS 2016. The Internet Society (Feb 2016)
42. Panchenko, A., Niessen, L., Zinnen, A., Engel, T.: Website fingerprinting in onion routing based anonymization networks. In: Chen, Y., Vaidya, J. (eds.) WPES 2011. pp. 103–114. ACM (Oct 2011). https://doi.org/10.1145/2046556.2046570
43. Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management (Aug 2010), version v0.34, https://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf
44. Pfitzmann, A., Köhntopp, M.: Anonymity, unobservability, and pseudonymity — A proposal for terminology. In: Federrath, H. (ed.) DPET 2000. LNCS, vol. 2009, pp. 1–9. Springer, Heidelberg (Jul 2001). https://doi.org/10.1007/3-540-44702-4_1
45. Pulls, T., Dahlberg, R.: Website fingerprinting with website oracles. PoPETs **2020**(1), 235–255 (Jan 2020). https://doi.org/10.2478/popets-2020-0013
46. Reed, M.G., Syverson, P.F., Goldschlag, D.M.: Proxies for anonymous routing. In: ACSAC 1996. pp. 95–104. IEEE Computer Society (1996). https://doi.org/10.1109/CSAC.1996.569678
47. Rényi, A.: On measures of entropy and information. In: Neyman, J. (ed.) Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1. vol. 4.1, pp. 547–561. University of California Press (Jan 1961)

48. Rimmer, V., Preuveneers, D., Juárez, M., van Goethem, T., Joosen, W.: Automated website fingerprinting through deep learning. In: NDSS 2018. The Internet Society (Feb 2018)
49. Rochet, F., Pereira, O.: Waterfilling: Balancing the tor network with maximum diversity. PoPETs **2017**(2), 4–22 (Apr 2017). https://doi.org/10.1515/popets-2017-0013
50. Rochet, F., Pereira, O.: Dropping on the edge: Flexibility and traffic confirmation in onion routing protocols. PoPETs **2018**(2), 27–46 (Apr 2018). https://doi.org/10.1515/popets-2018-0011
51. Serjantov, A., Danezis, G.: Towards an information theoretic metric for anonymity. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 41–53. Springer, Heidelberg (Apr 2002). https://doi.org/10.1007/3-540-36467-6_4
52. Sirinam, P., Imani, M., Juárez, M., Wright, M.: Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 1928–1943. ACM Press (Oct 2018). https://doi.org/10.1145/3243734.3243768
53. Standaert, F.X., Malkin, T., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 443–461. Springer, Heidelberg (Apr 2009). https://doi.org/10.1007/978-3-642-01001-9_26
54. Sun, Y., Edmundson, A., Vanbever, L., Li, O., Rexford, J., Chiang, M., Mittal, P.: RAPTOR: Routing attacks on privacy in tor. In: Jung, J., Holz, T. (eds.) USENIX Security 2015. pp. 271–286. USENIX Association (Aug 2015)
55. Syverson, P.F.: Why i'm not an entropist. In: Christianson, B., Malcolm, J.A., Matyas, V., Roe, M. (eds.) SPW 2009. LNCS, vol. 7028, pp. 231–239. Springer, Heidelberg (2009)
56. The23rd Raccoon: How I Learned to Stop Ph34ring NSA and Love the Base Rate Fallacy (2008), https://archives.seul.org/or/dev/Sep-2008/msg00016.html
57. Wagner, I., Eckhoff, D.: Technical privacy metrics: A systematic survey. ACM Computing Surveys **51**(3), 57:1–57:38 (Jun 2018). https://doi.org/10.1145/3168389
58. Wang, T., Cai, X., Nithyanand, R., Johnson, R., Goldberg, I.: Effective attacks and provable defenses for website fingerprinting. In: Fu, K., Jung, J. (eds.) USENIX Security 2014. pp. 143–157. USENIX Association (Aug 2014)
59. Wang, T., Goldberg, I.: Improved website fingerprinting on Tor. In: Sadeghi, A., Foresti, S. (eds.) WPES 2013. pp. 201–212. ACM (Nov 2013). https://doi.org/10.1145/2517840.2517851
60. Winter, P.: Towards a censorship analyser for Tor. In: Crandall, J.R., Wright, J. (eds.) FOCI 2013. USENIX Association, Washington, D.C. (Aug 2013)
61. Winter, P., Lindskog, S.: How the great firewall of China is blocking Tor. In: Dingledine, R., Wright, J. (eds.) FOCI 2012. USENIX Association, Bellevue, WA (Aug 2012)
62. Winter, P., Pulls, T., Fuss, J.: ScrambleSuit: A polymorphic network protocol to circumvent censorship. In: Sadeghi, A., Foresti, S. (eds.) WPES 2013. ACM (Nov 2013). https://doi.org/10.1145/2517840.2517856

A Example Scenario in Our Framework

Example 2 (Traffic analysis). Another example is represented by traffic analysis, in which a passive adversary captures ingress traffic (between the proxy and

guard) and egress traffic (between the exit node and the destination) and wishes to correlate the ingress traces with the egress traces. Such a threat model arises when an adversary controls the ISPs of several users and exit nodes. DeepCorr by Nasr *et al.* [37] is a good example of such an attack.

Our framework translates it to:

- S A sample from the set of bipartite graphs having $\mathcal{U}$, $\mathcal{D}$ as parts. $|\mathcal{U}| = |\mathcal{D}|$, and the random variable S selects uniformly a permutation. The bipartite graph is a graphical representation of such permutation.
- G State of the network after the setup phase ended and the traffic generated. It contains on all the circuits and the traffic traces they carried.
- V The adversary has information from two different parts of the network: the links, respectively, between the user and the entry node and from the exit node to the end destination. The first contains user's identity and ingress traffic, the latter destination's identity and egress traces.

goal The query "Which egress traffic can be associated to each ingress trace?"

- Z The training data used by the adversary's classifier.
- O A subset of $\mathcal{U} \times \mathcal{D}$, containing pairs (u_i, d_j) as a result of post-processing of the information returned by the classifier.

The DeepCorr classifier outputs a confusion matrix, where entry (i, j) scores how likely flows i and j belong together. Possible post-processing is a simple thresholding according to a parameter η. Note that with this post-processing, the output may end up inconsistent, as multiple egress traffic traces could be associated to the same ingress trace. To compare their results with RAPTOR by Sun *et al.* [54], they also considering an alternative post-processing, where for each ingress flow, the egress flow with the highest score is selected.

As in the previous example, authors use TPR and FPR as metrics to estimate the effectiveness of their attack, and *accuracy* in the comparison with RAPTOR.

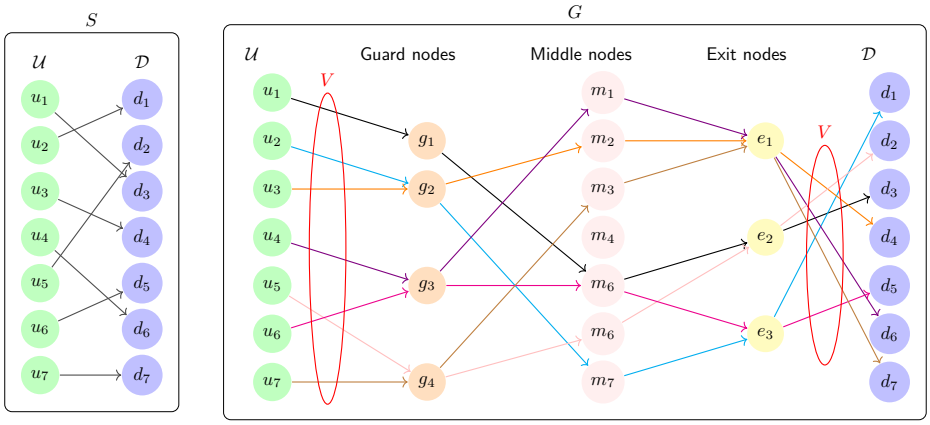
B Examples of Security Goals

Example 3 (Distinguishing goal). The adversary selects two websites, e.g. BBC or CCN, and the user picks one of them, without revealing it to the adversary. Then, the latter aims to discover which destination the user is visiting, assuming that each website is equally likely.

Formally, $\mathcal{U} = \{u\}, \mathcal{D} = \{d_1, d_2\}$ and the sample space of S is $\mathcal{U} \times \mathcal{D} = \{(u, d_1), (u, d_2)\}$, with a uniform probability. The adversary is presented with one of the two possible bipartite graphs they have previously chosen, and they are interested in identifying which one.

Example 4 (Decisional goal). Consider some state-controlled authority tasked with surveillance over citizens accessing censored websites over Tor (i.e. monitored websites), or a private company aiming to stop employees from using forbidden services (cloud, online games). The goal of such actors is to decide whether any observed traffic belongs to the censored category or not.

Fig. 5. Bipartite and multipartite graphs representing settings and random variables in DeepCorr (Example 2). In this setting, the adversary accesses information in two different parts of the network and their view V includes both ends of the circuits. Even in this case, though, edges in the bipartite graph representing the secret S still expand including more information such as the links guard-middle and middle-exit nodes.



For concreteness, assume 10 users are each accessing 100 destinations over Tor, the first 20 of which are monitored. So $\mathcal{U} = \{u_1, \ldots, u_{10}\}$ and $\mathcal{D} = \{d_1, \ldots, d_{100}\}$. If we further assume the users are each independently at random connecting to a single website in $\mathcal{D}$ then the distribution for S is fixed as well. The monitored websites form the target for the query, so $\mathtt{T} = \mathtt{T}_{\mathcal{D}} = \{d_1, \ldots, d_{20}\}$.

Here, false positives correspond to guess that some user visited a monitored website when, in fact, no user accessed any of them; conversely, false negatives correspond to missing that a monitored website was accessed. Depending on the setting, the adversary may want to minimise the probability of a specific type of wrong guess (or, equivalently, maximise the probability of a specific type of correct guess). As a consequence, they will process the information accordingly and the metrics will consider such imbalance. If the adversary is some state-authority, then we can assume they may be willing to have more false positives to minimise the false negatives. If the adversary is a private company, we can assume they may want to minimise the number of false positives instead.

Example 5 (Classification goal). The adversary targets a single user u and they want to determine which websites, from Alexa Top 500[1], the user is browsing using Tor. They are aware of the prior probability for each website in the list, previously estimated based on public available information.

Let $\mathcal{U} = \{u\}, \mathcal{D} = \{d_1, \ldots, d_{500}\}$ and the sample space of S be $\mathcal{U} \times \mathcal{D}$, with the estimated prior as probability distribution. The adversary is interested to identify the edges of the bipartite graph.

[1] `https://www.alexa.com/`, a service collecting browsing statistics and offering lists of most-visited websites.

The lack of preferences among outcomes, in this example, means that the adversary does not process the data in a way to minimise wrong guesses for a particular destination; instead, they aim to achieve, e.g, a better overall accuracy, while taking into consideration the estimated prior.

Example 6 (Computational goal). The adversary is interested in de-anonymising as many users as possible from a pool of 100, and they know that each of them visited a website from a pool of 100 different destinations (multiple users may have accessed the same one). No additional information Z is available to the adversary, so a uniform prior is assumed.

Let $\mathcal{U} = \{u_1, \ldots, u_{100}\}, \mathcal{D} = \{d_1, \ldots, d_{100}\}$ and the sample space of S be $\mathcal{U} \times \mathcal{D}$, with a uniform probability. The adversary is interested to know, for each user, the destination of the corresponding edge in the bipartite graph.

Since the adversary aims to maximise the number of de-anonymised users, the metric needs to consider outputs closer to the correct guess (i.e. a guess identifying all actual connections) as better than others: for example, a guess correctly identifying 90 destinations is preferred compared to an output with 60 correct user-destination edges.

C Selected Goals

Table 4 contains a list of commonly encountered goals.

D Information-Theoretic Notions

Rényi entropy is a generalization of Shannon entropy, defined as

$$\mathrm{H}_\alpha(X) = \frac{1}{1-\alpha} \log \sum_x P_X(x)^\alpha. \tag{2}$$

Selecting α "equal to" (apologies for sloppy notation) $0, 1, 2$, and ∞ we get, respectively: the max entropy, the Shannon entropy, the collision entropy, and the min entropy. Setting $\alpha > 1$ increases the emphasis on higher than average probability outcomes, which makes sense in the context of an adversary that can exploit e.g. guessing outcomes that are more than average probable.

As remarked above, the notion of entropy in the context of evaluating attacks makes sense only when combined with conditional entropy, mutual information, and information leakage. Defining a conditional Rényi entropy [47] has proved to be tricky. A conditional Rényi entropy should satisfy two inequalities that hold for conditional Shannon entropy, the monotonicity condition

$$\mathrm{H}_\alpha(X \mid Y) \leq \mathrm{H}_\alpha(X)$$

and the chain rule

$$\mathrm{H}_\alpha(X \mid Y) \geq \mathrm{H}_\alpha(XY) - \mathrm{H}_0(Y),$$

Table 4. List of goal examples.

Settings		Target $\mathtt{T}$	Description	Privacy notion [34]	Goal type
$\mathcal{U} = \{u\}$	$\deg(u) = 1$	$\mathtt{T}_{\mathcal{D}} = \mathcal{D}_u$	**1.** Single user visiting a single destination picked from a fixed set $\mathcal{D}_u$, goal is to identify which destination: closed world website fingerprinting.	$\mathsf{D}\overline{\mathsf{O}}$	Classification
	$\deg(u) > 1$	$\mathtt{T}_{\mathcal{D}} = \mathcal{D}_u$	**2.** Single user visiting multiple destinations picked from a fixed set $\mathcal{D}_u$, goal is to identify as many destinations as possible: multi-instance closed world website fingerprinting.	$\mathsf{D}\overline{\mathsf{O}}$	Computational
	$\deg(u) \geq 1$	$\mathtt{T}_{\mathcal{D}} \subsetneq \mathcal{D}_u$	**3.** Single user visiting one or more destinations, goal is to identify whether any destination belongs to a fixed set $\mathtt{T}_{\mathcal{D}}$: open world website fingerprinting.	$\mathsf{D}\overline{\mathsf{O}}$	Decisional
	$\deg(u) \geq 1$	$\mathtt{T}_{\mathcal{D}} \subsetneq \mathcal{D}_u$	**4.** Single user visiting one or more destinations, goal is to identify whether any of them belongs to a fixed set $\mathtt{T}_{\mathcal{D}}$ and, in case, which destination: hybrid website fingerprinting.	$\mathsf{D}\overline{\mathsf{O}}$	Classification
$\lvert\mathcal{U}\rvert > 1$	$\deg(u_i) = 1$	$\{\bar{u}\} \times \mathtt{T}_{\mathcal{D}}$, $\mathtt{T}_{\mathcal{D}} = \mathcal{D}_{\bar{u}}$	**5.** Multiple users, each of them visiting a single (possibly different) destination. User $\bar{u}$ is known to access a destination from a set $\mathcal{D}_{\bar{u}}$, goal is to identify which destination: noisy closed world fingerprinting.	$(\mathsf{UD})\overline{\mathsf{L}}$	Classification
	$\deg(u_i) = 1$	$\mathcal{U} \times \mathtt{T}_{\mathcal{D}}$, $\mathtt{T}_{\mathcal{D}} \subsetneq \mathcal{D}$	**6.** Multiple users, each of them visiting a single (possibly different) destination, goal is to identify which users are accessing specific destinations from a fixed set $\mathtt{T}_{\mathcal{D}}$.	$(\mathsf{UD})\overline{\mathsf{L}}$	Computational
	$\lvert\mathcal{D}\rvert > 1$, $\deg(u_i) = 1$	$\mathcal{U} \times \mathcal{D}$	**7.** Multiple users, each of them visiting a single (possibly different) destination, goal is to identify the destination of as many users as possible: traffic analysis.	$(\mathsf{UD})\overline{\mathsf{L}}$	Computational

Table 5. Some output-dependent metrics for different types of goals.

Metric	Definition	Description
Decisional:		
Precision	$\frac{TP}{TP+FP}$	The ratio of correct positive guesses over the total of positive outputs.
Recall	$\frac{TP}{TP+FN}$	The ratio of correct positive guesses over the total of positives.
Classification:		
Accuracy	$\frac{\sum_{i=j} c_{ij}}{\sum c_{ij}}$	$C = (c_{ij})$ is the confusion matrix. The ratio of correct guesses over the total.
Computational:		
Anonymity Set Size	$\#AS(X) \coloneqq \lvert X \rvert$	The number of users that can be linked to the traffic.

where X and Y are arbitrary stochastic variables. Several definitions have been proposed, but Fehr and Berens [23] showed that, among these, only the Arimoto definition [1] satisfies both the monotonicity rule and the chain condition. By this definition, $\mathrm{H}_\alpha(X \mid Y) = -\log R_\alpha(X|Y)$ where

$$\mathrm{R}_\alpha(X \mid Y) = \left(\sum_y \left(\sum_x P_{XY}(x,y)^\alpha \right)^{\frac{1}{\alpha}} \right)^{\frac{\alpha}{\alpha-1}}.$$

D.1 Input-dependent versus Output-dependent Metrics

The distinction we make between input-dependent and output-dependent metrics is one of convenience. Imagine a processing chain, starting with V and possible Z, deriving successively simpler representations until arriving at some $O_{\mathtt{q}}$ aligned to a specific query $\mathtt{q}$. There may be intermediate outputs along this chain: for example, the neural network used by DeepCorr [37] outputs a confusion matrix that is then transformed into a $\{0,1\}$-valued matrix.

It can be hard to place metrics along this input-output axis. *Anonymity set size* was proposed as a (computational) output-dependent metrics by Chaum [12], while Serjantov and Danezis [51] proposed an entropic version, *effective anonymity set size*. Actually, the latter is essentially equal to the conditional entropy of eq. (1). This conditional aspect does not seem to be picked up on elsewhere (cf. [57, ref [121] in Section 5.1.2]).

In some cases estimates of probability distributions are produced as outputs for the adversary. This is a potential source of confusion in the calculation of

entropy related functions. For example, Diaz *et al.* [16] write that "the attacker assigns a probability p_i", indicating that the attacker's output is a posterior distribution. However, the posterior *as output by an attacker* need not be the true posterior and it is certainly possible for an attacker to output vectors p_i (unrelated to the actual experiment) to result in almost any *degree of anonymity*. See also [57, ref [39] in Section 5.1.4].

Comparison. We now show simple examples of how input- and output-dependent metrics capture different aspects of the game and can change almost independently.

The settings are picked to highlight this behaviour and represent the basic cases: a single user picking one of two destinations d_1, d_2, which can result in two different traffic patterns t_1, t_2 and different probabilities associated to each of the patterns. Based on the information gathered from their guard node, the adversary outputs $\mathrm{Dec}(t) = \arg\max_d (p(S = d|V = t))$.

In the first scenario, destination d_1 causes t_1 with probability $\frac{2}{3}$ and t_2 with probability $\frac{1}{3}$; for destination d_2, the probabilities are, respectively, $\frac{2}{5}$ and $\frac{3}{5}$. Let's assume the destination is chosen with uniform probability ($\mathrm{H}(S) = 1$), and the adversary to observe the guard node, so their view contains information on the traffic traces. Then the view V has two possible outcomes (t_1, t_2) with associated probabilities equal to, respectively, $\frac{8}{15}$ and $\frac{7}{15}$, hence $\mathrm{H}(S \mid V) = 0.948$ and $\mathrm{I}(S; V) = 0.052$. The adversary computes $p(S = d_1|V = t_1) = \frac{5}{8}$ and $p(S = d_1|V = t_2) = \frac{5}{14}$, so $\mathrm{Dec}(t_1) = d_1$ and $\mathrm{Dec}(t_2) = d_2$, resulting in an accuracy of $\frac{19}{30} = 0.633$.

Let's assume a similar scenario, but $p(S = d_1) = \frac{2}{7}$, $p(S = d_2) = \frac{5}{7}$. Now, $\mathrm{H}(S) = 0.863$, $\mathrm{H}(S \mid V) = 0.821$ and $\mathrm{I}(S; V) = 0.042$: even if the user's prior probability is quite skewed, there is little information shared by S and V. The adversary computes $p(S = d_1|V = t_1) = \frac{2}{5}, p(S = d_1|V = t_2) = \frac{5}{11}$, hence $\mathrm{Dec}(t_1) = \mathrm{Dec}(t_2) = d_2$. The adversary is correct if and only if $S = d_2$, so the accuracy is $\frac{5}{7} = 0.714$.

For the third scenario, let's now assume a prior probability over d_1, d_2 — $\mathrm{H}(S) = 1$, and $p(V = t_1|S = d_1) = \frac{7}{8}, p(V = t_1|S = d_2) = \frac{1}{5}$. Entropy and mutual information are now $\mathrm{H}(S \mid V) = 0.637, \mathrm{I}(S; V) = 0.363$, while the accuracy is equal to $\frac{67}{80} = 0.838$, due to $p(S = d_1|V = t_1) = \frac{35}{43}, p(S = d_1|V = t_2) = \frac{5}{37}$.

Lastly, if also the prior probability is not uniform, e.g. $p(S = d_1) = \frac{5}{6}$ and $p(S = d_2) = \frac{1}{6}$, hence $\mathrm{H}(S) = 0.650$, we obtain $\mathrm{H}(S \mid V) = 0.432$ and $\mathrm{I}(S; V) = 0.218$. The conditional probabilities are $p(S = d_1|V = t_1) = \frac{175}{183}, p(S = d_1|V = t_2) = \frac{25}{57}$, so the accuracy is then $\frac{69}{80} = 0.863$.

This simple comparison on input metrics and output metrics shows that similar values of mutual information do not always correspond to similar values of accuracy or vice versa: the first and second scenarios result, respectively, in $\mathrm{I}(S; V) = 0.052, \mathrm{I}(S; V) = 0.042$, but the accuracies of the adversary are 0.633 and 0.714. Similarly, in the last two scenarios, the accuracies are 0.838 and 0.863, but the mutual information values equal to 0.363 and 0.218.