UNIVERSITÄT
DES
SAARLANDES

Saarland University

Faculty of Mathematics and Computer Science

Quantitative Anonymity Guarantees for Tor

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

von
Sebastian Wilhelm Ludwig Meiser

Saarbrücken,
Februar 2016

Day of Colloquium	23. November 2016
Dean of the faculty	Prof. Dr. Frank-Olaf Schreyer
Chair of the Committee	Prof. Dr. Christian Rossow
Reviewers	Prof. Dr. Michael Backes
	Prof. Aniket Kate, Ph.D.
	Dr. George Danezis
Academic Assistant	Dr. Robert Künnemann

Zusammenfassung

In dieser Arbeit präsentieren wir eine neue Methode, um beweisbar sichere Anonymitätsgarantien für Protokolle zur anonymen Kommunikation zu geben und veranschaulichen die Anwendung dieser Methode anhand des Tor Protokolls. Zu diesem Zweck präsentieren wir zunächst das AnoA Framework, welches dazu dient, die Anonymität von Protokollen zur anonymen Kommunikation zu quantifizieren. Der Angreifer, der in AnoA betrachtet wird, ist modular konfigurierbar. Wir zeigen, wie das Tor Protokoll in einer abstrakten Weise analysiert werden kann und greifen hierbei auf eine existierende Formalisierung von Tor im Universal Composibility (UC) Framework zurück. Wir beweisen, dass unsere abstrakten Betrachtungen sichere Garantien ergeben. Darüber hinaus entwickeln wir effizient berechenbare und beweisbar sichere Formeln, mit denen sich die Sicherheit von Tor gegenüber von Angreifern, die entweder durch Besitz eigener Tor Server oder durch eine Überwachung der Kommunikation, auf eine passive Art versuchen, die Anonymität zu brechen. Letztlich führen wir eine umfassende Analyse des Tor Netzwerks durch und berechnen die Anonymität, die Tor gegen eine Vielzahl solcher Angreifer zur Verfügung stellt, basierend auf den technischen Eigenschaften von Tor's Pfadsuchalgorithmus (und einigen Alternativen).

Abstract

In this thesis, we present a methodology to compute sound anonymity guarantees for anonymous communication protocols, which we apply to the Tor protocol. To this end, we present AnoA, a formal framework for quantifying the anonymity of anonymous communication protocols against modularly describable adversaries. We show how the Tor protocol can be analyzed in an abstract way by utilizing an existing formalization of Tor in the universal composability (UC) framework and prove that this analysis is sound. Moreover, we derive efficiently computable, sound anonymity guarantees for Tor against eavesdropping adversaries that may control Tor servers or wiretap communications between parties. For known adversaries that perform perfect traffic correlations we show that our bounds are tight. Finally, we perform an extensive analysis of the Tor network against such adversaries in which we calculate the anonymity of Tor when using Tor's current path selection, thereby considering many real-world properties of the path selection algorithm, as well as when using one of several variants, against a variety of eavesdropping adversaries that can control Tor nodes (depending on various properties) or wiretap the communication between them.

Acknowledgments

Sufficiently acknowledging all people that positively influenced me during my doctoral studies is certainly a hard task. I will try to keep these words brief and hope to not offend in doing so. I want to thank my supervisor Michael Backes for his support, for the various explicit and implicit lessons in these past years and for giving me sufficient freedom in my research agenda. I obviously want to thank all my colleagues with which I worked together in these past years, foremost Esfandiar Mohammadi; I enjoyed our discussions, the invention of our grand board game, and, being able to call you a friend. Moreover, I want to thank Aniket Kate for his undying optimism regarding our projects, as well as for the foresight in striving for such types of guarantees. I want to thank Dominique Unruh for teaching me the art of designing definitions and proofs and Tim Ruffing for all helpful discussions.

Concerning the line of research started by this thesis I want to thank all students and internship students that I had the pleasure to supervise (in almost alphabetical order): Markus Bauer, who both helped in fixing the implementation of MATor and who as his Bachelor's thesis developed a MATor plugin for the Tor browser, Simon Koch, who successfully battled parenthesis to perform a large-scale data collection for our network-infrastructure adversaries, Tahleen Rahman who had the bravery to delve into the depths of Tor's C-implementation and who extended this dissertation in her Master's thesis by modeling adversaries that add new Tor nodes to the consensus, Marcin Slowik who significantly contributed to more than one refactoring of MATor and with whom I had helpful discussions about the nature and generality of adversarial advantages, Lukas Wedeking, who as his Bachelor's thesis analyzed the impact of adversaries on Tor's hidden services on basis of this dissertation, and finally Marta Wrzola and Dawid Gawel who in their internship contributed to the final refactoring of MATor.

I also want to thank the Cognition and Action research group of Prof. Dirk Wentura, as well as Prof. Christian Frings for giving me insights into psychology and pragmatism when solving problems, as well as for their encouraging discussions concerning my doctoral studies.

Finally, I want to thank my family for their support and my friends for the countless hours of stress-relieving pen-and-paper games.

For Science.

Background of this Dissertation

This dissertation is based on three peer-reviewed papers that have been accepted at influential and well-ranked conferences in the field of computer security, in each of which the author contributed as one of the main authors. In addition, the author was the main author of other peer-reviewed papers in the field of cryptography and information security [11, 10] and one of the main authors of a paper slightly related to the foundation of this thesis [16]. However, this dissertation focuses on the following four papers, as this selection defines a novel and very promising line of work. The author considers a thorough and consistent presentation that partially improves upon the original works to be significantly more useful to the research community than less consistent presentation that additionally covers the papers mentioned above.

- In AnoA [13] (published at CSF'13), as well as in the extended technical report and the journal version (to be published in a special issue of the Journal of Privacy and Confidentiality), the author contributed significantly to the central notions of the framework, whereas a preliminary and abstract analysis of the Tor network (which is not included here) was mainly done by Praveen Manoharan. The central notion of *IND-CDP* is influenced by the papers "Computational Differential Privacy" [81] of Mironov et. al, and "Secrecy without Perfect Randomness: Cryptography with (Bounded) Weak Sources" [16] in which the author also contributed as one of the main authors. The adaptations of the UC protocol to the AnoA framework were performed by Esfandiar Mohammadi. Praveen and Esfandiar also worked on the UC-realization of differential-privacy style properties, which is a central building block for AnoA.

- In the paper "(Nothing Else) MATor(s)" [14] (published at CCS'14), the author contributed as one of the main authors. Together with Esfandiar Mohammadi, he investigated how the actual Tor client implements Tor's current path selection algorithm, as well as the impact of these subtle choices on Tor's anonymity. Moreover, they investigated the impact of TCP ports and derived sound formulas for calculating anonymity guarantees for Tor. The author additionally contributed to the implementation of *MATor*, a tool used for calculating Tor's anonymity that forms the foundation of all implementations considered for the evaluations in this thesis.

- In "Your choice MATor(s)" [12] (published at PETS'16), the author was the main author and driving force. The author contributed by defining a novel, observation based analysis for calculating the anonymity guarantees, that is significantly more precise than the heuristics used before. The necessary modifications and the partial re-implementation of the analysis tool MATor, however, was done by Marcin Slowik, who worked under the author's direct supervision and the evaluation was designed and performed by both authors.

- Finally, in a paper we are currently working on that is targeted at analyzing the impact of network-level adversaries on Tor, the author contributed mainly to the theoretical foundations by defining a novel, observation based analyses of the calculation of our anonymity guarantees, together with the proofs on its correctness. The gathering of data was performed by Simon Koch, Tor's Internet topology was developed by the author together with Esfandiar Mohammadi and Christian Rossow.

As an additional effort for this thesis, the individual papers have been combined into one consistent analysis of Tor that includes a novel formalization of a simplified AnoA game for Tor against eavesdropping adversaries. To this end, the majority of proofs has been revised, the analysis tool MATor has been re-factored and we repeated our large-scale formal evaluation of Tor against eavesdroppers.

Contents

Chapter 1

Introduction

The Internet has evolved from a small communication network, used only by academic researchers and technology enthusiasts to the most culture-defining technology of the last century. Within few decades, the Internet has grown to a backbone of communication and trade, and even short-time Internet blackouts for individual companies result in economic damage estimated in ten thousands US dollars per minute of downtime. Moreover, with the development of smartphones and social networks, the Internet has become a major social factor in the life of the majority of people: cheap communication, almost at the speed of light, from one continent to another is affordable for billions of people and integrated into various entertainment systems. In addition to enhancing social communication and possibilities, the Internet has also introduced various previously unthinkable challenges to privacy. As recent revelations have shown, companies can gather vast quantities of information about their employees and customers, and countries can even gather and monitor data from every communication of any two citizens with each other and store this data for years, thus achieving eavesdropping powers previously only described in a science fiction dystopia.

These invasions of people's privacy have motivated millions of people to use anonymous communication (AC) protocols to anonymously communicate or browse the web. In all AC protocols, traffic is sent via anonymizing proxies, i.e., servers are trusted to relay the users' traffic as if it originated from them, without revealing the identity of the users. One of the most important protocols is the onion routing network Tor [42], a widely employed low-latency anonymous communication service [109] used by more than 2 million users every day. To provide anonymity, Tor routes a user's traffic through several anonymizing proxies. For each communication via Tor the trust is distributed over three anonymizing proxies (also called nodes), instead of one proxy, which are chosen from around 6500 volunteer nodes [110]. Using these anonymizing proxies, Tor creates an anonymous channel for the user that ideally provides anonymity as long as not all three proxies are corrupted – an important strategy, since not all volunteers can be trusted [116]. In contrast to mix-nets, Tor allows for low-latency traffic, which is a strict requirement for comfortable web surfing.

Whenever AC protocols strive for low-latency traffic, so called *traffic-correlation attacks* become possible. Traffic correlation attacks allow to re-identify traffic even if the traffic is encrypted, decrypted, or re-encrypted, based on the patterns of the traffic. As traffic at different points in the network can be correlated, such attacks render Tor's ideal anonymity goals (partially) invalid and lead to the following central question from a user's perspective:

> How anonymous is this channel that Tor creates, i.e., how likely is it that an eavesdropper can deanonymize me?

Deriving the degree of a user's anonymity is challenging for such a complex system with a fluctuating set of 6500 nodes, where each node has different parameters, such as its entrusted bandwidth and a set of TCP ports it offers for a communication. Previous analyses are either empirical or abstract the Tor network by ignoring important real-life parameters of Tor, such as the path selection algorithm, the varying entrusted bandwidth of different Tor nodes, or the user's requested ports. So far, there is no methodology for performing an accurate, formally grounded analysis of the anonymity provided by Tor.

1.1 Contribution

In this thesis we present a full formal methodology for quantifying the anonymity of an AC protocol against eavesdropping adversaries and we apply this methodology to the popular AC protocol Tor. Our contribution is manifold. We first propose a novel perspective on anonymity, by defining several prominent notions of anonymity as indistinguishability-based notions. Based on this intuition, we introduce *AnoA*, a formal framework for deriving quantitative anonymity guarantees for the prominent anonymity notions *sender anonymity*, *recipient anonymity* and *relationship anonymity.* We then apply AnoA to Tor, utilizing an existing formalization of Tor in the universal composability framework (UC) and show that AnoA's central anonymity guarantee is compatible with UC. Consequently, anonymity guarantees presented for an idealized version of Tor carry over to its cryptographic instantiation, and thus, under reasonable assumptions, to its implementation. We then analyze the possible impact that an eavesdropper can have on Tor's anonymity and define the concept of adversarial *observations* and their impact on the respective anonymity notions. We show that a fairly abstract, observation-based definition of Tor suffices for giving correct guarantees by relating this abstract notion to Tor's UC formalization. We derive a precise method for calculating the impact of such an observation-based eavesdropper and show that for *fixed adversaries*, i.e., for adversaries with a fixed set of eavesdropping points, this calculation is precise up to a negligible factor. We show how to efficiently give an approximative guarantee for a whole class of adversaries and prove that these guarantees are sound. Finally, we utilize the presented definitions and abstractions to perform the largest formal evaluation of the Tor network so far. We analyze the impact of eavesdroppers on Tor's anonymity, which we model both in terms of compromised Tor nodes and in terms of overly curious network infrastructure.

In what follows we describe our key contributions in slightly more detail.

AnoA – a framework for quantifying anonymity We present AnoA, a framework for quantifying anonymity of autonomous communication (AC) protocols. AnoA defines anonymity as a challenge-response game between a probabilistic polynomial time (PPT) Turing machine A and a challenger machine Ch that simulates the AC protocol. AnoA is parametric in the targeted anonymity notion which it represents as a function that depending on a challenge bit selects a scenario, e.g., for sender anonymity, the anonymity function selects one of two possible senders. The goal of the adversary is to predict the challenge bit by distinguishing the possible scenarios:

Informal Definition 1.1.1 (Reduction of Anonymity (Intuitive))**.** *Let α be an anonymity function. An adversary* A *that outputs either* 0 *or* 1 *reduces the anonymity of a protocol* Π *by at most* ε *and* δ*, if the probability of the adversary to distinguish the challenge bit used by the anonymity function is bounded by* ε *and* δ*:*

$$\begin{aligned} &\texttt{Pr}\left[0 = \langle \mathrm{A} | \textsc{Ch}(\Pi, \alpha(0)) \rangle\right] \\ \leq\ & e^{\varepsilon}\ \texttt{Pr}\left[0 = \langle \mathrm{A} | \textsc{Ch}(\Pi, \alpha(1)) \rangle\right] + \delta, \end{aligned}$$

where $z = \langle X|Y \rangle$ describes the output of the machine X after interacting with the machine Y.

For deriving guarantees against adversaries with different observational capabilities and for defining the number of Tor nodes the adversary can compromise and the number of wires it can tap, we cluster adversaries into so called *adversary classes* that describe the adversaries' capabilities. We explore and define a set of conditions for an adversary class such that we achieve *single-challenge reducibility* (SCR), i.e., guarantees for one challenge suffice to generically derive guarantees for an arbitrary number of challenges. To simplify subsequent analyses even further, we present a versatile, constructive definition of adversary classes, which we coin *Plug'n'Play* adversary classes that by construction satisfy the conditions for SCR.

The anonymity impact of eavesdropping adversaries on Tor For analyzing Tor, we instantiate the protocol Π in our AnoA definition with a formalization of Tor in the universal composability (UC) framework, as presented by Backes et.al. [15]. To counter trivial attacks by the adversary, we slightly modify the formalization by the use of some wrapper machines for the environment. We analyze Tor's anonymity and to simplify subsequent analyses we show that considering one message per session suffices for quantifying anonymity.

We define an abstract notion of the *observations* an adversary can make when eavesdropping on Tor communication. Building upon this notion of observations we define a significantly more abstract and simpler anonymity game in which we replace the UC formalization of Tor with a simple observation function. We subsequently show that this simplification is sound and tight for any eavesdropping adversary: we show how to transform any adversary against Tor's UC formalization into an adversary against the observation function and vice versa.

Calculating precise formal guarantees We present a methodology for calculating the impact of passive adversaries on Tor, based on our notions of observations. We define different types of eavesdropping adversaries, based on the novel concept of *budget-adversaries*, which describe all adversaries that can compromise Tor entities and connections between them, based on a budget and a cost function that assigns costs to every compromisable entity and connection. We show a generic way for calculating guarantees for any eavesdropping adversary that compromises a given set of Tor nodes and connections and prove these calculations sound and tight up to a negligible factor. Given a budget adversary class, we then show how to calculate guarantees in a rather efficient manner and show that these guarantees are sound.

Evaluating Tor's anonymity Finally, we show the applicability of our methodology by performing a large-scale evaluation of Tor's anonymity against eavesdroppers. Our evaluation considers both Tor's current path selection algorithm and interesting variants thereof. Moreover, we analyze the impact of a large number of budget-adversaries, including adversaries that corrupt a specific number of Tor nodes, adversaries that corrupt Tor nodes with at most a given amount of bandwidth in total, adversaries that corrupt all Tor nodes in specific countries and adversaries that corrupt Tor nodes based on their monthly monetary upkeep costs. Moreover, we present a second analysis of Tor against network infrastructural adversaries, i.e., adversaries that compromise part of the Internet infrastructure, which allows them to eavesdrop on connections between nodes.

Remark 1.1.1. *This thesis contains work published in several international peer-reviewed top conferences (CSF'13 [13], CCS'14 [14], ACNS'15 [16], and PETS'16 [12]). A description of the AnoA framework that is more along the lines of the presentation in this thesis is also to appear in the journal of privacy and confidentiality (JPC).*

1.2 Overview

In this section we present the intuition behind key concepts of the thesis together with several informal definitions of these concepts. To find a common basis for discussion, we first describe the type of anonymity properties we aim at and isolate it from other possible interpretations of anonymity and privacy. Based on this intuition of anonymity, we describe our novel indistinguishability-based methodology for defining anonymity, in which an adversary tries to distinguish between so called anonymity scenarios. For the three commonly considered anonymity notions sender anonymity, recipient anonymity, and relationship anonymity, we define indistinguishability-based variants and compare our definitions with similar definitions in the literature. Furthermore, we give a brief overview over the anonymous communication protocol Tor and describe what an eavesdropping adversary can observe. We show how to cast these observations into formulas that describe the reduction of anonymity imposed by the eavesdropper. Moreover, we introduce our evaluation of Tor against several such eavesdroppers.

1.2.1 Anonymity and Privacy

In the literature, the concepts of anonymity and privacy are often not clearly distinguished. Some works consider anonymity an essential ingredient for privacy (how can you protect your privacy, if anyone can trace your actions?), others consider privacy a necessity for anonymity (how can you be anonymous if you leak personal information?). In this section, we briefly discuss the concepts of privacy and anonymity in the context of for the context of communication between two participants and choose one of many possible interpretations of anonymity that we will use for the rest of this thesis. We regard anonymous communication protocols as basic building blocks for constructing privacy preserving applications and protocols. Hence, we strive for a definition of anonymity that (a) is independent of the application in which the anonymous communication protocol is used, (b) is strong enough to hold independent of the context in which it is used, and (c) matches our intuition of anonymity. Consequently, we consider anonymity and privacy to be separate properties which we define as follows.

Informal Definition 1.2.1 (Privacy)**.** *Privacy is a property of the content of messages (that might be publicly available). We distinguish between* attribute disclosure*, where the messages leak information about certain attributes of their writer, e.g., age, gender, or personal interests, and* identity disclosure*, where messages leak information about the physical identity of their writer. We consider the end-points of the communication to be independent of this content and thus irrelevant for an analysis.*

Ideal privacy means that the content of messages does not leak information about their writer.

Informal Definition 1.2.2 (Anonymity)**.** *Anonymity is a property of a communication channel that specifies whether or not the end-points of this channel can be identified. We consider the transferred information to be independent from the end-points and thus irrelevant for an analysis.*

Ideal anonymity means that the communication can be considered an idealized anonymous channel, where the sender unobservably sends a message via an unobservable direct connection to a trusted third party that, in turn, relays the message via an unobservable direct connection to the recipient.

Remark 1.2.1. *The combination of anonymous communication and content-based information (which includes information about the structure of content) is non-trivial: existing attacks on anonymity include so called* website fingerprinting, *in which the structure of even encrypted and otherwise anonymous communication is used to identify which website a user visits. Similarly, by observing the stream of communication, one can typically easily distinguish between a user that receives a small website and a user that streams a video, simply by the shape and sheer amount of transferred information.*

These definitions cover only a subset of the notions and applications considered in the literature, but allow us to clearly describe the properties we strive to achieve. The integration of both notions is non-trivial and requires application-specific analyses. In this thesis, we solemnly focus on anonymity, which we consider to be an important and necessary building block for a wide variety of applications.

1.2.2 Intuition: Indistinguishability Based Anonymity

We consider three common notions of communication anonymity α in this paper: *sender anonymity* ($\alpha = \alpha_{\mathsf{SA}}$, i.e., determine who is sending a message), *recipient anonymity* ($\alpha = \alpha_{\mathsf{RA}}$, i.e., determine to whom a message is being sent), and *relationship anonymity* ($\alpha = \alpha_{\mathsf{REL}}$, i.e., determine a correlation between sender and recipient). Each of these notions is defined as the (in-)ability of an adversary to distinguish two *scenarios* that differ in their involved senders and recipients. This follows the established concept of indistinguishability-based definitions in cryptography (e.g., IND-CCA secure encryption): The adversary can initially corrupt Tor nodes and tap wires between them (including wires from senders to Tor nodes and from Tor nodes to recipients). Then, one of these two scenarios is selected at random, a Tor circuit is created for this scenario, and the adversary is then allowed to make observations for this circuit depending on the set of corrupted nodes. The adversary knows the set-up of both scenarios, makes its observations, and it then has to decide which scenario it currently observes.

In the following we present all three anonymity notions as so called *anonymity scenarios* that should be hard to distinguish. For every scenario we present an intuitive description, as well as an abstract definition. We compare the indistinguishability-based notions based on scenarios with commonly considered notions of anonymity from the literature.

Informal Definition 1.2.3 (Indistinguishability-based Anonymity)**.** *We say that a protocol* Π *satisfies* α *up to* δ *for* $\alpha \in$ {*sender anonymity, recipient anonymity, relationship anonymity*} *against an adversary* A*, if* A *cannot distinguish between the respective anonymity scenarios with an advantage more than* δ*.*

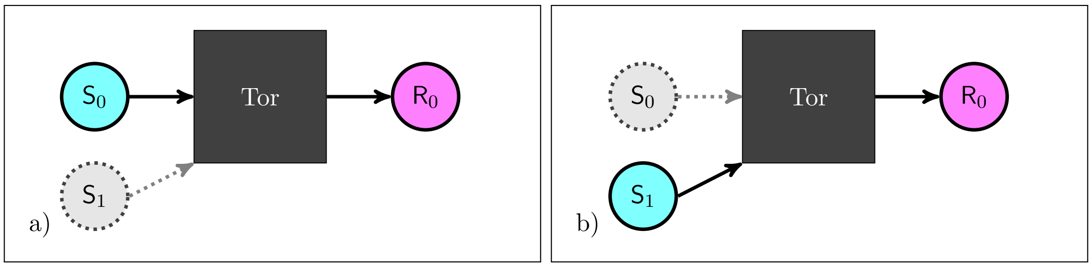


Figure 1.1: Sender anonymity scenarios. Either S_0 communicates to R_0 (a), or S_1 communicates to R_0 (b).

Sender Anonymity

Sender anonymity models that the sender of a message is well hidden and cannot be easily discovered.

Sender Anonymity Scenarios For *sender anonymity*, we compare the communication from S_0 to R_0 to the communication from another sender S_1 to the same recipient R_0. We say, Tor satisfies sender anonymity, if even a malicious recipient R_0 cannot distinguish between the two senders. Consequently, the two scenarios differ in the sender, but share the same recipient R_0. In addition to its observations from corrupted nodes and communication between corrupted nodes, the adversary is allowed to observe the recipient R_0 and should be able to distinguish if the communication originates at S_0 or at S_1. We refer to Fig. 1.1 for a graphical presentation of the scenarios for sender anonymity.

Sender Anonymity in the Literature The notion of sender anonymity is introduced in [93] as follows:

> Anonymity of a subject from an adversary's perspective means that the adversary cannot sufficiently identify the subject within a set of subjects, the anonymity set.

In comparison to our indistinguishability-based notion of sender anonymity, this classical definition requires the adversary to directly identify a sender, i.e., to compute a sender with sufficient probability, where the anonymity set intuitively describes how hidden a sender is. Naturally, our anonymity notion is stronger than the classical definition above.

Informal Lemma 1.2.1 (Sender anonymity comparison 1)**.** *For all protocols* Π *over a (finite) user space* $\mathcal{S}$ *it holds that if* Π *provides* δ*-sender anonymity against any* `PPT` *adversary* A *(of certain strength), as in Informal Definition 1.2.3, then classically, no similar* `PPT` *adversary* B *(with the same strength) can compute the sender with probability more than* δ *for any anonymity set* $N \subseteq \mathcal{S}$*.*

Proof Idea. We show the contraposition of the lemma: an adversary B that breaks classical sender anonymity, can be used to distinguish between the anonymity scenarios. We construct an adversary A against Informal Definition 1.2.3 by running B and by comparing the output of B with the possible senders of our scenarios. If B outputs one of those senders, A outputs the respective scenario. A has at least the success probability in the indistinguishability game as B in the sender anonymity game. Since A has no additional information, it is of similar strength as B. □

In the converse direction, we lose a factor of $\frac{1}{N}$ in the reduction, where N is the size of the anonymity set. If an AC protocol Π provides δ-sender anonymity, we classically can only guarantee $(\delta \cdot N)$-sender anonymity for Π.

Informal Lemma 1.2.2 (Sender anonymity comparison 2)**.** *For all protocols* Π *over a (finite) user space* $\mathcal{S}$ *it holds that if no* `PPT` *adversary* B *(of certain strength) can compute the sender* S *out of an anonymity set* N *with probability more than* δ*, then* Π *provides* $(N \cdot \delta)$*-sender anonymity as in Informal Definition 1.2.3 against any* `PPT` *adversary* A *(with the same strength).*

Proof Idea. We show the contraposition of the lemma: an adversary A that breaks sender anonymity as in Informal Definition 1.2.3, can be used to break classical sender anonymity. We construct an adversary B against classical sender anonymity by running A and outputting S_0 if A outputs 0 and S_1 if A outputs 1. If the senders from the anonymity scenarios coincide with the randomly chosen sender from the classical notion, the reduction works (with probability $\frac{1}{N}$ the sender is S_0; with probability $\frac{1}{N}$ the sender is S_1). Thus, B has an advantage of $\frac{\delta}{|N|}$ of outputting the sender, if A has an advantage of δ in distinguishing the scenarios. □

Recipient Anonymity

Recipient anonymity models that the recipient of a message is well hidden and cannot be determined easily.

Recipient Anonymity Scenarios For *recipient anonymity* we compare the communication from S_0 to R_0 to the communication from the same sender S_0 to another recipient R_1. We say Tor satisfies recipient anonymity if even an adversary observing all outgoing traffic of S_0 cannot distinguish between the two recipients. Similarly to sender anonymity, the two scenarios differ in the recipient, but share the same sender S_0. We refer to Fig. 1.2 for a graphical presentation of the scenarios for recipient anonymity.

Recipient Anonymity in the Literature Recipient anonymity can analogously to sender anonymity be defined via anonymity sets and the (in)ability of an adversary to determine the real recipient within that set. Our observations and informal lemmas from sender anonymity equally apply to recipient anonymity by the same arguments.

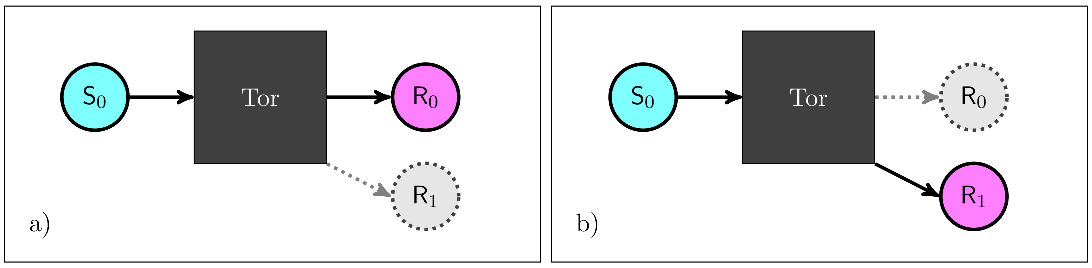


Figure 1.2: Recipient anonymity scenarios. The sender S_0 either communicates with R_0 (a) or with R_1 (b).

Relationship Anonymity

Relationship anonymity models that it is hard to determine the relationship between senders and recipients, i.e., which sender communicates to which recipient. For any given message, it should be hard to determine the sender of that message and the recipient to which it is sent.

Relationship Anonymity Scenarios Capturing the absence of correlations to define relationship anonymity is more involved. We consider both an additional sender S_1 and an additional recipient R_1: The first relationship anonymity scenario considers the two cases that S_0 communicates with R_0 and that S_1 communicates with R_1; the second scenario considers the two cases that S_0 communicates with R_1 and that S_1 communicates with R_0. After the scenario has been selected, one of the two described cases for this scenario is chosen uniformly at random, then a Tor circuit is created for this case, and the adversary can make its observations for this circuit. We refer to Fig. 1.3 for a graphical presentation of the scenarios for relationship anonymity.

Relationship Anonymity in the Literature We find that, for the notion of relationship anonymity, many of the interpretations from the literature are not accurate. In their Mixnet analysis, Shmatikov and Wang [101] define relationship anonymity as 'hiding the fact that party A is communicating with party B'. Feigenbaum et al. [47] also take the same position in their analysis of the Tor network. However, in the presence of such a powerful adversary, as considered in this work, these previous notions collapse to recipient anonymity since they assume knowledge of the potential senders of some message.

We consider the notion of relationship anonymity as defined in [93]: the anonymity set for a message m comprises the tuples of possible senders and recipients; the adversary wins by determining which tuple belongs to m. However, adopting this notion directly is not possible: an adversary that gains partial information (e.g. by finding out the sender of a communication), also breaks this relationship anonymity game, as the sender-recipient pairs are no longer equally likely. Therefore we think that our indistinguishability-based approach constitutes a better definition of relationship ano-

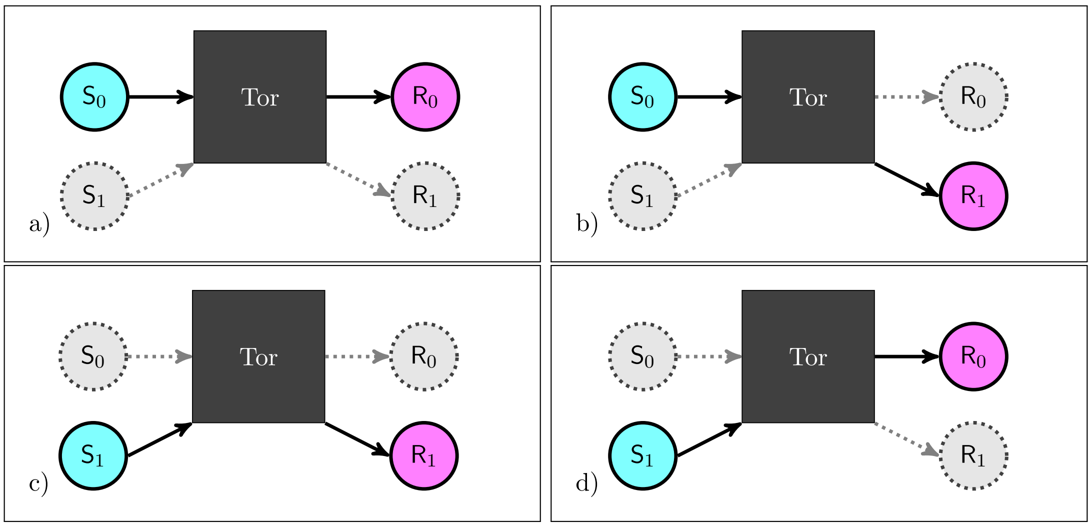


Figure 1.3: Relationship anonymity scenarios. On the left side are the scenarios in which S_0 communicates with R_0 (a) or S_1 communicates with R_1 (c). These should be hard to distinguish from the scenarios on the right side, where S_0 communicates with R_1 (b) or S_1 communicates with R_0 (d).

nymity, as the adversary needs to uncover both sender and recipient in order to learn the correlation between them and, thus, to break anonymity.

1.2.3 The Tor Protocol

When communicating over the Internet, the IP address of both the sender of a message and the recipient of a message are visible in plain text. The recipient of a message needs to be readable to allow the message to actually be forwarded to its destination. The sender is included to allow the recipient to correlate the received packets and to send its reply to the correct IP address. Both IP addresses are visible to every entity along the path taken through the Internet, which allows a multitude of entities to store the communication history of users and to build profiles about their personal and political opinions, thus invading their privacy.

To hide this correlation between a sender's IP address and a recipient's IP address, many use *proxies*, i.e., servers that relay messages from the sender to the recipient. Such proxies (also called *relays*, or in the case of Tor, *nodes*) hide any direct connection between the sender and the recipient of a communication. The Tor network [42, 109] consists of a large number of around 6,500 such anonymizing proxies (in October 2015), solely contributed by volunteers and usable without fee. Using any such proxy, however, places a significant trust on the provider of the proxy that, by definition, sees all incoming and outgoing traffic of any user and can create profiles on its own. Tor achieves anonymity by the three main principles of a large variety in users, the distribution of trust over three proxies per communication and the unpredictability of communication. We first describe the technical design decisions of Tor that are inspired by these three principles and then present a few of the available statistics about the Tor network.

To increase the variety of Tor users, Tor tries to please a large number of users. The majority of users care about efficiency and comfort at least as much as about privacy. Consequently, Tor aims at providing low-latency communication with sufficient bandwidth suitable for browsing the web, thus significantly increasing the acceptance of the Tor network among common privacy-aware users. The distribution of trust over three Tor nodes (instead of one anonymizing proxy) significantly improves the anonymity against malicious Tor nodes. Every communication is relayed over a so called *circuit* consisting of three Tor nodes: the guard node, the middle node and the exit node. Each node can only observe and relay traffic from its predecessor to its successor and all traffic is encrypted. Moreover, a circuit is constructed in a telescopic way: the sender first establishes a secure channel with the guard node, then, using this secure channel establishes a second secure channel to the middle node, and finally, using both secure channels he establishes a third secure channel to the exit node. By way of construction the choice of exit node is hidden from the guard node and vice versa. If a circuit was created, using the Tor node $\mathsf{n_g}$ as guard node, $\mathsf{n_m}$ as middle node and $\mathsf{n_x}$ as exit node, the sender has exchanged keys $k_{\mathsf{n_g}}$, $k_{\mathsf{n_m}}$, and $k_{\mathsf{n_x}}$ with these nodes respectively. For sending a message m to a recipient R, the sender creates a layered encryption of the message and sends $\textsc{Enc}(k_{\mathsf{n_g}}, \textsc{Enc}(k_{\mathsf{n_m}}, \textsc{Enc}(k_{\mathsf{n_x}}, (m, \mathsf{R}))))$ to the guard node. The guard node $\mathsf{n_g}$, using $k_{\mathsf{n_g}}$ removes one layer of encryption and sends $\textsc{Enc}(k_{\mathsf{n_m}}, \textsc{Enc}(k_{\mathsf{n_x}}, (m, \mathsf{R})))$ to the middle node. The middle node $\mathsf{n_m}$, using $k_{\mathsf{n_m}}$ removes another layer of encryption and sends $\textsc{Enc}(k_{\mathsf{n_x}}, (m, \mathsf{R}))$ to the exit node. The exit node $\mathsf{n_x}$ using the key $k_{\mathsf{n_x}}$ removes the last layer of encryption and sends m to the specified recipient R. Because of the layered encryptions that are "peeled" while sending the message, this type of routing is called *onion routing*. We refer to Fig. 1.4 for a graphical representation of this technique. Ideally, as long as at least one of the Tor nodes within a circuit is honest and as long as there are many users that create Tor circuits, the communication is anonymous, as every information aside from the predecessor and the successor of a node is hidden by means of cryptography, with the exception that the exit node can observe the message itself. To achieve confidentiality of messages, the sender and the recipient have to exchange keys and send encrypted messages via Tor. Tor clients choose fresh Tor nodes for every Tor circuit via a randomized selection method that (ideally) uses the same distribution over Tor nodes for all senders and recipients. Thus, controlling a small number of, say, $k = 65$ malicious Tor nodes out of the set of $n = 6500$ Tor nodes, an adversary can only succeed if three of its Tor nodes were chosen for the same circuit, resulting in a deanonymization chance of roughly $\left(\frac{k}{n}\right)^3 = 0.0001\%$.

The ideal description of Tor presented so far is almost identical to the actual implementation of the Tor protocol, with a few highly significant differences that are essential for Tor's anonymity.

- The fact that Tor communication aims at low-latency communication allows for so-called *traffic correlation attacks*. For example, assume that Alice uses Tor to

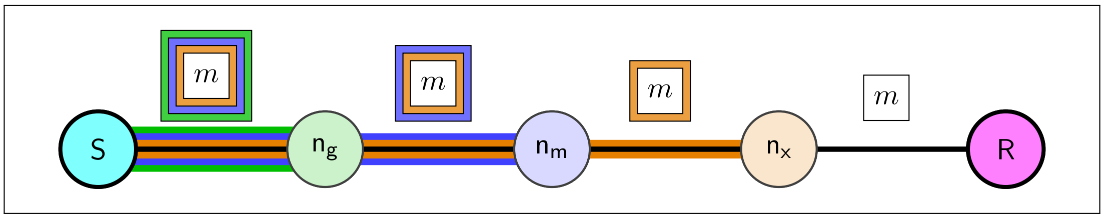


Figure 1.4: Channels for layered (onion) encryption. The sender S sends the message m to the recipient R, using $\mathsf{n_g}$ as the guard node, $\mathsf{n_m}$ as the middle node and $\mathsf{n_x}$ as the exit node of a Tor circuit. The encryption here is displayed via encapsulating the message in a rectangle. The color of the rectangle corresponds to the color of the entity with which the sender exchanged the respective cryptographic key.

anonymously communicate to Bob. If Eve can observe traffic from Alice into the Tor network, then Eve can note down the number of packets that Alice sends, as well as the delay between those packets. If furthermore Eve observes, at roughly the same time, traffic from the Tor network to Bob, she can compare the number of packets and their delays with her notes and in some cases re-identify Alices traffic. By applying a traffic correlation attack, it suffices for Eve to control the guard node and the exit node of a communication. Moreover, if Bob controls the guard node of Alice, he can apply traffic correlation attacks to deanonymize Alice, since he observes all traffic Alice sends to him.

- The fact that Tor aims to provide sufficient bandwidth has lead to very influential bandwidth-weights on the random choice over Tor nodes. The Tor client chooses Tor nodes not uniformly but by a weighted random choice, where the weight directly corresponds to (some measurement of) the entrusted bandwidth of a Tor node. Consequently, a node that provides significantly more bandwidth than other nodes will have a higher probability to be chosen.

- Since not everyone who provides a Tor node wants this node to be used as an exit node, and since the guard position is considered more significant than the other two positions, Tor does not allow every node at every position, which further restricts the choices and thus modifies the probabilities: Nodes are assigned flags, such as `Exit`, if they can be used as exit nodes `Guard`, if they can be used as guard nodes, or `Stable` if they are considered stable enough for Tor circuits that are required to work for more than a few minutes.

As these differences between the idealistic view of Tor and the actual implementation heavily impact Tor's anonymity, we require a formalization of Tor that is precise enough to capture all presented subtleties. The anonymity analyses presented in this thesis consider all mentioned properties correctly.

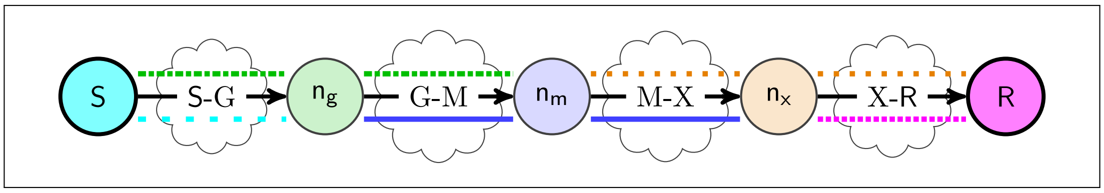


Figure 1.5: Observation points of an eavesdropper.

1.2.4 From Observations of an Eavesdropper to Anonymity Guarantees

When using Tor, one of the biggest open research questions is the impact of malicious Tor nodes and the impact of eavesdroppers on the anonymity of users. There are two main strategies for adversaries: performing aggressive active attacks, such as denial-of-service attacks on Tor nodes or exploiting vulnerabilities in the protocol to watermark traffic and performing undetectable, passive attacks, such as contributing a seemingly honest Tor node but simultaneously trying to deanonymize users or to wiretap communication between senders, Tor nodes and recipients. Aggressive active attacks, although devastating, are easily detectable and typically are also fixable via some modifications of the Tor protocol. Consequently, in this work we focus on the second type of adversaries: passive eavesdroppers that try to undetectably deanonymize Tor users, either by contributing Tor nodes or by wiretapping communication in parts of the Internet.

For any given Tor circuit that a sender S creates to communicate to a recipient R, such an eavesdropper can sit at many different positions: it can control any of the three nodes of a circuit $(\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x})$ and eavesdrop on traffic between the sender and the guard node (S-G), between the guard node and the middle node (G-M), between the middle node and the exit node (M-X) or between the exit node and the recipient (X-R). We refer to Figure 1.5 for a graphical overview over these positions.

At any observation point at which the adversary can observe traffic, it can also observe the protocol entities that communicate, e.g., a malicious guard node can observe the sender and the middle node of a communication. Such observations alone can suffice for breaking anonymity in terms of the anonymity scenarios from Informal Definition 1.2.3. In this thesis we show that observations of this granularity suffice for calculating sound quantitative anonymity guarantees against eavesdropping adversaries. To this end, we first enumerate all possible relevant observations that an adversary could make for any given circuit, i.e., all combinations of possible senders, guard nodes, middle nodes, exit nodes and recipients. For every such observation and for every anonymity scenario we calculate the probability that the considered adversary can make this observation, where the probability is based on the random choices of the sender that creates a circuit. More precisely, given any observation o, any circuit $\mathsf{C} = (\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R})$ and an adversary A, we check whether A would make the observation o for the circuit C. We then calculate the probability for every circuit C to be created by the sender S of the anonymity scenario and accumulate the probabilities of all circuits for which the adversary makes the same observation. Finally, we compare the probability for each such observation depending on the anonymity scenario and accumulate all differences in probabilities to yield an anonymity guarantee. We prove that for every passive eavesdropper A these calculations indeed yield a precise characterization of the advantage of A in distinguishing between the respective anonymity scenarios.

Proving that the technique presented here results in sound (and for fixed adversaries also tight) guarantees is the main technical contribution of this thesis. As the purpose of this overview is to give an intuitive description of the targeted problem, we will not discuss the strategy of our proofs here. Instead, we refer the interested reader to Chapter 2 for our formal definitions of anonymity, to Chapter 3 for our definitions of observations and their impact on Tor and to Chapter 4, where we derive a methodology for calculating anonymity guarantees for eavesdroppers based on their observations and show this methodology to be sound and, for fixed adversaries, tight.

1.2.5 Defining Categories of Eavesdroppers

We aim to apply the analysis sketched in the previous section to perform a large-scale evaluation of the Tor network against passive eavesdroppers. To this end, we define a variety of such eavesdroppers, depending on their attack vectors, and embed them into our formal framework. In the following we sketch all considered adversaries, grouped into *curious Tor node adversaries* that compromise or simply own a number of Tor nodes and *malicious network infrastructure adversaries* that monitor Internet traffic on a large scale.

Curious Tor Node Adversaries

We model curious Tor node adversaries by defining sets of *budget adversaries* $\mathcal{A}_f^B$ that are parametric in a cost function f that assigns a cost to every Tor node and a budget B that limits the expenses of the adversary. Every adversary $\mathrm{A} \in \mathcal{A}_f^B$ controls a set of Tor nodes s.t. the sum of the costs of all contained nodes is below the specified budget. We then instantiate this fairly general adversary to model the following adversaries.

k-collusion Adversary A k-collusion adversary is the most straight-forward instance of our budget adversary, where f assigns the cost 1 to every Tor node and allows for a budget of k compromised nodes. This adversary models collusions of up to k Tor nodes. Its worst-case instance typically compromises the Tor nodes with the highest bandwidth.

Predicate Adversary Given any predicate P on Tor nodes, such as whether the node runs a certain (vulnerable) version of the Tor software or whether it is within a specified geographic jurisdiction, a predicate adversary sets the cost of all nodes that satisfy the predicate to 0 and the costs of all nodes that do not satisfy the predicate to ∞. The budget $B = 1$ states that the adversary can only compromise nodes for which the predicate is satisfied. This adversary class directly specifies which nodes the adversary will compromise and can be used to define arbitrary *fixed* adversaries.

Bandwidth-constrained Adversary A bandwidth-constrained adversary can compromise arbitrary Tor nodes that contribute up to a certain, specified amount of bandwidth. This adversary models that a certain *percentage of Tor's bandwidth* is compromised. The cost function f assigns to each node a cost equal to the node's bandwidth and the budget B specifies the total compromised bandwidth.

Money-constrained Adversary The money-constrained adversary can compromise arbitrary Tor nodes, but has to pay the (estimated) monthly costs required to run them. This adversary models an adversary following a rational strategy that tries to maximize its deanonymization power while minimizing its costs. The cost function f assigns to each node a cost equal to the nodes monthly renting cost and the budget B directly specifies the adversary's monetary budget.

Malicious Network Infrastructure Adversaries

We model malicious network infrastructure as a predicate adversary that compromises the communication into the Tor network, out of the Tor network or between Tor nodes, but that does not compromise Tor nodes. The main difficulty in defining such types of adversaries lies in learning the topology of the Internet, which is a hard problem on its own.

Autonomous System(s) Adversary The autonomous system (AS) adversary compromises all communications between Tor nodes that run through one or more specified autonomous systems. This adversary describes the impact that any Internet service provider (or a collusion of several such companies) can have on Tor's anonymity.

Internet Exchange Point Adversary The Internet exchange point (IXP) adversary compromises all communications between Tor nodes that run through the specified IXP. Since IXP's typically do not show up in measurements of the Internet, we infer the existence of an IXP on a route by considering the autonomous systems and the physical location of the traversed subnets and assume that an IXP is present if two ASes that openly peer at the IXP are traversed subsequently and if furthermore the subnets are in geographic proximity of the IXP.

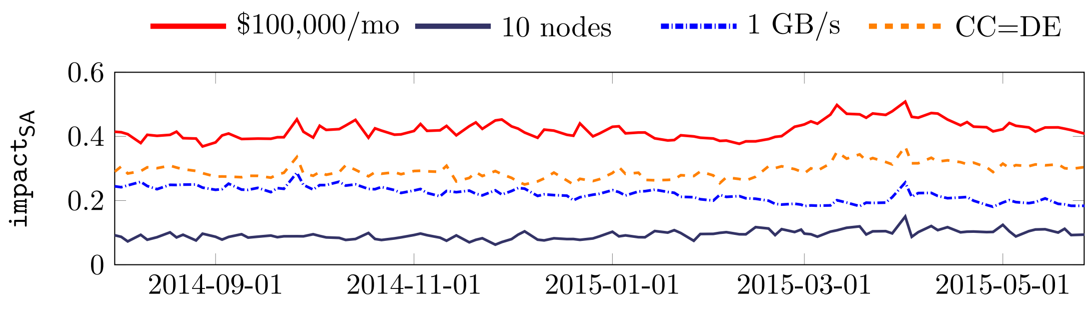


Figure 1.6: One of the results of our analysis: The graph shows the advantage of the respective adversary to distinguish between sender anonymity scenarios. The adversaries include (from left to right) an adversary that can compromise arbitrary Tor nodes, but has to pay for their monthly renting costs and that can spend up to 100,000 US dollars per month, an adversary that can compromise 10 arbitrary Tor nodes of its choice, an adversary that can compromise arbitrary Tor nodes with a total Bandwidth of at most 1 GB/s, and an adversary that compromises all Tor nodes in Germany.

Submarine Cable Adversary The submarine cable adversary compromises all communications between Tor nodes that run through the specified submarine cable, or that run through any cable that passes through the specified landing point. For inferring the usage of a submarine cable, we try to learn the geographic location of the traversed IP subnets and infer the usage of a cable by the usage of subnets that are close to two (different) landing points of the same cable, typically accompanied by a change of continents.

1.2.6 Evaluating Tor Against Eavesdroppers

For evaluating the Tor network we have performed several analyses over the course of around one year, from August 2014 until June 2015. In our evaluation we compare the impact of the adversaries from above on Tor's anonymity for several variants of Tor's path selection algorithm.

1.3 Related Work

Our field of anonymity research is an ever-changing field that has attracted a lot of attention in the last decades. We partition the related work into the categories *anonymous communication and (onion) routing protocols*, *attacks and countermeasures*, and *anonymity analyses*. We begin by discussing a few selected anonymous communication protocols and then focus on onion routing and, most prominently, the widely used onion routing protocol Tor. After presenting these protocols we discuss a few types of attacks that are conceptually hard to avoid when using anonymous communication protocols, especially when using low-latency AC protocols for interactively browsing the web. Finally we review a number of existing analyses on AC protocols, compare them with this work and group them into *formal analyses* and *empirical analyses.* We end the section by discussing why we did not take an entropy-based approach to anonymity.

In what follows we discuss each category in more detail.

1.3.1 Anonymous Communication Protocols

Over the past decade, many protocols for anonymous communication have been proposed. Two approaches, both based on fundamental work by Chaum in the 1980's are mixnets and dining-cryptographer nets (DC-nets). AC protocols based on the dining cryptographers problem, called DC-nets [33] achieve impressive guarantees, but rely on a vast communicational overhead. Mix-nets [34], consist of proxies that receive traffic from different senders, wait until a sufficient amount of traffic has been sent and then try to pertube this traffic to hide the identity of the senders. Specific protocols aim to overcome the inefficiency of these original proposals. Examples include Herbivore [53], which is based on DC nets, Dissent, which is based on DC nets with mixing [36], which combines the ideas of DC nets with mix nets and even the cute idea of having an AC protocol imitate a public transport system to anonymously add and remove information to a bus-like message [20].

A different, but particularly successful methodology to provide anonymous communication is the concept of onion routing[94, 55]. Onion routing protocols allow for low-latency anonymous communication. Many onion routing approaches have been proposed, e.g., Tarzan [49], a peer-to-peer variant of onion routing, Salsa [88] that tries to help the node selection if the users only have incomplete knowledge of the set of nodes, and PIR-Tor [82], that adds private information retrieval techniques to such variants of onion routing to achieve a better scalability. Variants of onion routing protocols either focus on high performance [74, 35], or on very strong variants of anonymity [72]. Danezis et al. [37] explore onion routing on the basis of a social network, where friends act as relays.

The most popular onion routing protocol today, Tor [42], currently is used by around two million users every day [110]. Tor's success inspired many Tor related scientific papers that propose slight variations, present vulnerabilities and attacks and that try to analyze Tor's anonymity. Since the focus of this thesis lies in analyzing Tor's anonymity against a specific type of attacks, we now discuss papers reasoning about Tor in slightly more detail.

The Tor literature is rich on proposals for new path selection algorithms. Some propose to increase the anonymity of the users [1, 45, 70, 14, 71, 60] by reducing the attack vectors of an adversary that controls part of the Internet infrastructure or part of the Tor network. Others propose to increase the performance (i.e., expected latency and throughput) of the Tor network [80, 72, 66, 108, 84, 4, 8, 115, 2, 57, 3, 5].

However, although most of these proposals come with a specific evaluation highlighting the performance impact or the anonymity impact of the proposed change (rarely both), their analyses are incomparable. The rich literature on variants of Tor motivates us to provide a framework for objectively quantifying and comparing the anonymity impact of the different considered adversaries (e.g., adversaries that compromise Tor nodes or adversaries that compromise parts of the Internet infrastructure) against different variants.

1.3.2 Practical Attacks and Countermeasures

The field of anonymity research has discovered many strong attacks against anonymity protocols and subsequently tried to prevent them via countermeasures.

A very successful type of attacks is called *traffic correlation attacks* [77, 86, 102, 87, 44, 52, 31, 30, 83, 117, 69]. Applying a traffic correlation attack, the adversary can re-identify traffic it observed at other points in the network, even if the traffic has been encrypted, re-encrypted or decrypted in between by analyzing the patterns, timings and other features of the traffic that are preserved by the cryptographic operations applied in between. As traffic correlation attacks are not specific to Tor's implementation, usage or protocol specification, but that are practically unavoidable without making Tor as it is unusable, we capture them in our formalization.

There also is a rich literature on other, more specific attacks, out of which we present only a small selection of interesting attacks. Borisov et. al show that denial of service (DoS) attacks against Tor can also impact anonymity by increasing the adversary's attack surface [22]. The revelation of protocol vulnerabilities that allow cheap DoS attacks suggest that this might even be a successful strategy [17]; Jansen et al. present a way to anonymously host DoS attacks against powerful Tor nodes [68]. Similarly, Bauer et. al [18] leverage a (fixed) vulnerability that allowed Tor nodes to significantly cheat when reporting their bandwidth and thus to increase their impact on Tor. The attacks that abuse such vulnerabilities typically are both devastating to anonymity and detectable by the Tor network. Consequently, we do not focus on these protocol-specific (and fixable) attacks against Tor in this thesis. However, revelations of vulnerabilities typically result in the need to patch the Tor protocol, which motivates us to provide a reusable and easily configurable framework for analyzing Tor (and other AC protocols).

Adversaries can sometimes learn the so called fingerprint of a website, i.e., the amount of traffic it sends and the number of requests it creates. This leads to an attack in which an adversary can quite trivially break recipient anonymity of a low-latency AC protocol by identifying the recipient via its fingerprint, even if the traffic is encrypted. The literature on fingerprinting attacks also considers such content-based attacks against onion routing in general and Tor in particular [56, 91, 114].

1.3.3 Anonymity Analyses

Pfitzmann and Hansen [93] develop a consistent terminology for various relevant anonymity notions; however, their definitions lack formalism. Nevertheless, these informal definitions form the basis of almost all recent anonymity analysis, and we also adopt their terminology to validate the anonymity definitions in AnoA.

There has been a substantial amount of previous work [107, 39, 98, 100, 78, 58, 101, 38, 46, 47, 51, 6, 48] on analyzing the anonymity provided by various AC protocols such as dining cryptographers network (DC-net) [33], Crowds [95], mix network (Mixnet) [34], and onion routing (e.g., Tor) [50, 61, 94]. Previous frameworks such as [63] only guarantee anonymity for a symbolic abstraction of the AC protocol, not for its cryptographic realization. Moreover, while some existing works like [48] consider an adversary with access to *a priori* probabilities for the behavior of users, analyses struggle with adversaries that possess arbitrary auxiliary information about user behavior.

There have been analyses which focus on a particular AC protocol, such as [98, 38, 101, 51, 73] for Mixnet, [21, 6] for DC-net, [39, 100] for Crowds, [97] for Shadow, [105] for Dissent, and [107, 78, 46, 47, 70, 48] for onion routing. Most of these study a particular anonymity property in a particular scenario and are not flexible enough to cover the emerging system-level attacks on the various AC protocols. Some of the existing frameworks and analyses focus on changes in entropy [39, 98, 101, 38], which allows, but also requires, to explicitly specify the adversary's initial knowledge. We refer the readers to [48, Sec. 5] for a detailed survey up to 2012.

Formal Treatment of Anonymity Notably, there has been work on exploring metrics related to the multiplicative factor in our anonymity definitions: Andres et al. [6] distinguish multiplicative leakage and additive leakage (however, as alternatives, not in combination with each other). Their approach is focused on concurrent systems and applied to the dining cryptographers protocol. Moreover, Shmatikov [100] defines the notions of "probable innocence" and "possible innocence" – in terms of our metric, a multiplicative factor quantifies the strength of "probable innocence", while ensuring "possible innocence", whereas the distinguishing events rule out "possible innocence".

A relatively recent result [48] among these by Feigenbaum, Johnson and Syverson models the OR protocol in a simplified black-box abstraction, and studies a notion of relationship anonymity which is slightly different from ours: here the adversary wishes to identify the destination of a user's message. As discussed in Section 1.2.2, this relationship anonymity notion is slightly weaker than ours. Moreover, their model is not flexible enough to extend to other system-level scenarios such as fingerprinting attacks [92, 25, 44].

Building on the work of Hevia and Micciancio [61], Gelernter and Herzberg published, concurrently with our AnoA framework, an expressive framework that extends the work of Hevia and Micciancio with active adversaries that adaptively send inputs [50]. They apply their methodology to obtain an impossibility result for a strong combination of sender anonymity and unobservability, which they coin "ultimate anonymity". However, as in the work of Hevia and Micciancio, they only consider qualitative anonymity notions (i.e., protocols that only allow a negligible chance of de-anonymizing users), which does not allow them to quantify the anonymity loss in low-latency anonymous communication networks, such as Tor. Moreover, they do not provide a mechanism, such as the adversary classes presented in this work, to flexibly relax the strength of the adversary.

These pieces of work, however, abstract the Tor network by ignoring characteristics of Tor that significantly influence a user's anonymity, such as the path selection algorithm, the varying entrusted bandwidth of different Tor nodes, or the user's requested ports. All these works assume that the path selection algorithm chooses nodes uniformly at random (which Tor does not) and, hence, do not provide rigorous guarantees. Moreover, the formalizations of Tor [50, 48] ignore subtle, yet potentially influencial, differences in adversarial observations whenever different senders or recipients have an impact the probabilities of the selected Tor circuits, e.g., because they use specific parameters for Tor, or even alternative path selection algorithms.

Empirical Studies and Simulations Empirical studies and simulations are a very successful method for generating insights into the strengths and weaknesses of an anonymous communication protocol. In contrast to abstract theoretical analyses, these empirical works often are close to the practical implementation of the protocol in question and thus provide realistic results. However, the are, by nature, empirical and incomplete. For us, these works provide an inspiring motivation for formalizations and provably sound analyses.

This second line of work models Tor's anonymity relevant characteristics more accurately [69, 113, 19, 67, 29, 65, 85], e.g., they explicitly model Tor's path selection. Most works in this area focus on simulating Tor rather than on performing live experiments on Tor, as this might lead to privacy violations (c.f. [75] for a discussion). Johnson et al. [69] present a simulation of the Tor network, based on a probabilistic (bandwidth-based) adversary that compromises a certain percentage of Tor's bandwidth, which the adversary has to split into entry bandwidth and exit bandwidth. This strategy, however, neglects the case that a relay that offers its bandwidth as entry node will with significant probability be chosen as a middle node (i.e., the second proxy) or as an exit node (i.e., the third proxy) and can still learn some information.

The work by Wacek et al. [113] goes into a similar direction. They construct a slightly smaller model of the Tor network, on which they perform extensive simulations, using the real Tor client code, to evaluate the impact of path selection algorithms on Tor's performance and anonymity. As both analyses are based on simulations, they, in contrast to our work, do not yield provable bounds on the degree of a user's anonymity.

In the category of empirical analyses without rigorous anonymity guarantees, Murdoch and Watson [85] present an analysis of proposed path selection algorithms against (bandwidth-based) adversaries that can inject malicious nodes into the Tor network, subject to a specific adversarial budget. Their work inspired the formalization of our budget adversary, with the difference that our adversary compromises existing nodes instead of adding new nodes.[1] Other works strive to analyze Tor against network-level adversaries, which we consider a highly interesting, yet orthogonal, problem. In this area, Johnson et al.[70] and later Jaggard et al. [64] propose path selection adaptations based on network trust to reduce the impact of network adversaries. Wacek et al. [113] analyze the impact of path selection algorithms on anonymity and performance by simulating a significant fraction of the Tor network, and then they analyze the anonymity of various path selection algorithms against (AS-level) network adversaries. The amount of analyses based on simulations and measurements further underlines the importance of a rigorous approach for quantitatively assessing the anonymity of Tor's path selection algorithm and comparing it against alternative variants.

Some empirical works [19, 65, 99] provide practical tools for simulating the Tor protocol. We consider this a very helpful contribution to the community as it allows for rerunning experiments and for extending analyses in a comparable way. With our implementation of *MATor*, we strive to also provide a usable analysis tool for Tor that is, however, based on our formal analysis instead of empirical measurements.

Relation to Entropy-based Anonymity Notions Entropy-based analyses [111, 32, 24] use well-established anonymity notions to argue about the impact of an adversary. Hemal, Gregoire and Goldberg [59] argue that an attack-based view might be more favorable in some situations. They consider an adversary that has a certain bandwidth budget and tries to infiltrate as many paths as possible. As Paul Syverson portrays it in a well-written article [106], entropy is useful, but cannot easily capture some of the most important concepts we look for when analyzing anonymity. The anonymity guarantee we present should mainly depend on the hardness of an adversary to overcome the AC protocol and not present an average-case result that bears only a small importance to some unlucky users. Moreover, it should not be based on the number of other users in the system if they actually do not significantly impact the anonymity of individual users. Insights and thoughts like these actually lead us to develop AnoA and to focus on worst-case indistinguishability guarantees.

To avoid giving too pessimistic guarantees, we parametrize the adversary by its capabilities, its knowledge and its view of the world, thus reaching a balance between worst-case guarantees against an adversary with almost arbitrary prior knowledge and realistic assumptions and relaxations of such an adversary by explicitly reducing its knowledge.

Our indistinguishability-based notion reasons about the cryptographic implementation of Tor. For such cryptographic systems with their computational security guarantees, entropy-based notions, including notions that define the *effective size* of an anonymity set [98, 39], are not directly applicable. A relation between entropy-based

[1]An method for quantifying the impact of an adversary that adds new nodes to the network based on our AnoA framework and out notion of observations is developed by Tahleen Rahman as part of her Master's thesis.

notions and cryptographic notions might be possible along the lines of [9] that establishes a tight correspondence between the information-theoretic capacity of channels, their abstract description and finally their cryptographic instantiations. We plan to investigate this approach in the context of more comprehensive systems such as Tor in future work.

Chapter 2

AnoA

In this chapter we introduce AnoA, a formal framework for quantifying anonymity guarantees for anonymous communication (AC) protocols. We strive for a framework that is defined independent of the AC protocol we analyze. Moreover, we aim for definitions that match our intuition of anonymity scenarios, as discussed in Section 1.2.2. The framework should be parametric in the anonymity notion and should also be able to formalize arbitrary efficient adversaries.

To this end, we define the AnoA framework as a challenge-response game between an adversary machine and a challenge machine. The challenge machine simulates the anonymity scenarios (parametric in the anonymity notion). Finally, we define the concept of adversary classes for restricting adversaries and present a few formal results that make our upcoming analyses easier: We show that for some adversary classes analyzing the protocol for one challenge suffices for deriving results for multiple challenges. As this property comes from reducing adversaries that initialize several challenges to adversaries that only initialize one challenge, we coin it *single-challenge reducibility.*

Note that even though this thesis focuses on analyzing the anonymity of Tor against eavesdroppers, all results from this chapter are applicable to arbitrary AC protocols and other types of adversaries as well.

2.1 Notation

We consider the following sets of entities: a set of senders $\mathcal{S}$, a set of recipients $\mathcal{R}$ and a set of Tor nodes $\mathcal{N}$. Subsequently, we group senders, recipients and Tor nodes in the set of all relevant Tor entities, i.e., entities that can send and/or receive messages and write $\mathcal{E} = \mathcal{N} \cup \mathcal{S} \cup \mathcal{R}$.

2.2 Full AnoA Game

We formalize anonymity as a challenge-response game, in which the adversary has to distinguish two scenarios. Formally, an *anonymity notion* is a function α that receives as inputs two senders S_0 and S_1, two recipients R_0 and R_1, and a so-called challenge bit b. It then selects one sender and one recipient based on the challenge bit and the considered anonymity notion. This selection can be probabilistic and for every individual challenge the anonymity function can keep state. The anonymity of a protocol is then defined as a bound on the *adversary's advantage* in guessing the challenge bit.

Game-based anonymity definition The definition of the AnoA challenger $\textsc{Ch}$ is the main building block for the definition of the adversary's advantage in AnoA as a challenge-response game. In the AnoA framework, the challenger receives as input a description of the protocol Π to be analyzed, an anonymity notion α, a bound γ on the permitted challenge-messages (see-below), and the challenge bit b. It then simulates the protocol for the sender-recipient scenario selected by α. The adversary interacts with the challenger in order to determine which scenario is being simulated. The adversary knows all inputs to the challenger up to an uncertainty of one bit (the challenge bit b). We now describe the challenger $\textsc{Ch}$ in detail and refer to Figure 2.1 for an algorithmic definition of $\textsc{Ch}$.

The AnoA challenger.

- As described above, the challenger $\textsc{Ch}$ expects as inputs a description of the protocol Π to be analyzed, the anonymity notion α, a bound γ on the permitted challenge-messages, and the challenge bit b. The challenger keeps track of active sessions $\textsc{Sessions}$ and the states of all challenges in a list States consisting of γ entries that initially are $\texttt{FRESH}$.

- *Challenge-messages:* Upon receiving a message $\texttt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, m, \Psi) \in \mathcal{S}^2 \times \mathcal{R}^2 \times \{0,1\}^* \times \mathbb{N}$, the challenger checks whether the challenge index is allowed ($\Psi \leq \gamma$) and whether the challenge state allows further challenge messages with this tag ($\mathsf{States}[\Psi] \neq \texttt{OVER}$). If any of these checks fails, the challenger outputs an error symbol $\perp$ to the adversary. If they succeed, $\textsc{Ch}$ computes the anonymity notion α on $(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$, the challenge state $\mathsf{States}[\Psi]$ and the challenge bit b and obtains a sender-recipient pair $(\mathsf{S}^*, \mathsf{R}^*) \in \{\mathsf{S}_0, \mathsf{S}_1\} \times \{\mathsf{R}_0, \mathsf{R}_1\}$, as well as an updated challenge state state'. The challenger first stores the updated challenge state and then simulates the protocol where S^* sends the message m to R^*. We abbreviate this using the subroutine $\textsc{SimulateProtocol}(\Pi, \mathsf{S}^*, m, \mathsf{R}^*, (\textsc{Ch}, \Psi))$ in Figure 2.1. For protocols that use sessions: If no session ID $\textsc{sid}$ exists in $\textsc{Sessions}$ for the challenge Ψ, the challenger creates a new session $\textsc{sid} \leftarrow \mathbb{N}$ and stores it in $\textsc{Sessions}$ together with the challenge tag Ψ. Otherwise, it retrieves the existing session ID. Via the existing or freshly created session the challenger then sends the message m from S^* to R^* using Π.[1]

- *Input-messages:* Upon receiving a message $\texttt{Input} = (\mathsf{S}, \mathsf{R}, m, \mathsf{z}) \in \mathcal{S} \times \mathcal{R} \times \{0,1\}^* \times \mathbb{N}$, the challenger directly calls the subroutine $\textsc{SimulateProtocol}(\Pi, \mathsf{S}, \mathsf{R}, m, (\mathrm{A}, \mathsf{z}))$, as described above, with the difference that fresh sessions are created for every nonce z. These sessions are by definition disjoint from the sessions created for challenge messages. Input-messages, hence, capture additional information the adversary may have about the communication contents in the protocol.

Based on the challenge-response game introduced above, we can now define impact on anonymity of an adversary that may compromise protocol parties and connections between them, if the protocol description allows that. We also call this anonymity impact the *reduction of anonymity* imposed by the adversary. Our definition is motivated by similar definitions presented under different contexts, e.g. differential indistinguishability [16] for cryptography with imperfect randomness, and differential privacy [43] for privacy in statistical databases. Formally, we say that a protocol Π provides $(\alpha, \gamma, \epsilon, \delta)$-*IND-CDP* if the (in)distinguishability of the random variables defined by $\langle \mathrm{A} | \textsc{Ch}(\Pi, \alpha, \gamma, 0) \rangle$ and $\langle \mathrm{A} | \textsc{Ch}(\Pi, \alpha, \gamma, 1) \rangle$ can be bounded multiplicatively and additively be ε and δ. Note that we write $\langle A(x) | B(y) \rangle$ for the output of a machine A on input x when interacting with a machine B on input y and consequently write $z = \langle A(x) | B(y) \rangle$ for the event that said output equals z.

Definition 2.2.1 (Adversary's Advantage; Reduction of Anonymity)**.** *Let α be an anonymity function, $\gamma \in \mathbb{N}$. Then, the* adversary's advantage *of an adversary* A *against a protocol* Π *for γ challenges of the anonymity notion α is bounded by ε and δ, with $\varepsilon \geq 0, 0 \leq \delta \leq 1$, if for all sufficiently large $\eta \in \mathbb{N}$, we have*

$$\begin{aligned} & \texttt{Pr}\left[0 = \langle \mathrm{A}(1^\eta) | \textsc{Ch}(\Pi, \alpha, \gamma, 0) \rangle\right] \\ \leq \; & e^\varepsilon \; \texttt{Pr}\left[0 = \langle \mathrm{A}(1^\eta) | \textsc{Ch}(\Pi, \alpha, \gamma, 1) \rangle\right] + \delta. \end{aligned}$$

We say that a protocol Π exhibits a reduction of anonymity *of at most (ε, δ) under γ challenges (formally: Π is $(\alpha, \gamma, \epsilon, \delta)$-IND-CDP) against a class $\mathcal{A}$ of adversaries if the adversary's advantage of all probabilistic polynomial-time adversaries* $\mathrm{A} \in \mathcal{A}$ *against* Π *for γ challenges of the anonymity notion α is bounded by ε and δ.*

Anonymity Function The anonymity guarantee we provide in AnoA is parametric in the function $\alpha : \{0,1\}^* \times \mathcal{S}^2 \times \mathcal{R}^2 \times \{0,1\} \to \{0,1\}^* \times \mathcal{S} \times \mathcal{R}$ describing the anonymity notion. This function encodes the specific anonymity notion we analyze, e.g. sender anonymity, recipient anonymity or relationship anonymity. The main purpose of the anonymity function α is to validate challenges issued by the adversary for the

[1]The semantics of the simulation and the session handling depends on the protocol. For Tor, the sessions correspond to Tor circuits.

```
**Adaptive AnoA Challenger** $\textsc{Ch}(\Pi, \alpha, \gamma, b)$

**Initialize the Game**
  Set $\textsc{Sessions} := \emptyset, \mathsf{States} := [\mathtt{FRESH}, \mathtt{FRESH}, \ldots, \mathtt{FRESH}]$ ($\gamma$ entries, one per challenge).

**Upon message** $\mathtt{Input} = (\mathsf{S}, \mathsf{R}, m, \mathsf{z})$
  $\textsc{SimulateProtocol}(\Pi, \mathsf{S}, \mathsf{R}, m, (\mathrm{A}, \mathsf{z}))$

**Upon message** $\mathtt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, m, \Psi)$
  **if** $\Psi \notin \{1, \ldots, \gamma\} \vee \mathsf{States}[\Psi] = \mathtt{OVER}$ **then**
    Output $\perp$.
  **else**
    Compute $(\mathsf{state}', \mathsf{S}^*, \mathsf{R}^*) \leftarrow \alpha(\mathsf{States}[\Psi], \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$
    Set $\mathsf{States}[\Psi] := \mathsf{state}'$.
    $\textsc{SimulateProtocol}(\Pi, \mathsf{S}^*, \mathsf{R}^*, m, (\textsc{Ch}, \Psi))$
  **end if**

**Upon message** $(m, \textsc{sid})$ **from** $\Pi$
  **if** $(\mathsf{ID}, \textsc{sid}) \in \textsc{Sessions}$ for some value $\mathsf{ID}$ **then**
    Send $\mathtt{Observations} = (m, \mathsf{ID})$ to A.
  **end if**

**Subroutine SimulateProtocol**$(\Pi, \mathsf{S}, \mathsf{R}, m, \mathsf{ID})$**, where** $\mathsf{ID} = (\textsc{Ch}, \Psi)$ **or** $\mathsf{ID} = (\mathrm{A}, \mathsf{z})$
  **if** $(\mathsf{ID}, \textsc{sid}) \in \textsc{Sessions}$ for some value $\textsc{sid}$ **then**
    Retrieve this $\textsc{sid}$
  **else**
    Set $\textsc{sid} \leftarrow \mathbb{N}$
    Store $(\mathsf{ID}, \textsc{sid})$ in $\textsc{Sessions}$
  **end if**
  Run $\Pi$ on $(\mathsf{S}, \mathsf{R}, m, \textsc{sid})$. If the protocol supports sessions, use session $\textsc{sid}$ (or create a fresh one, if no session $\textsc{sid}$ exists). The precise semantics obviously depends on the protocol $\Pi$.
```

Figure 2.1: The full AnoA challenger.

anonymity notion represented by α and to choose one of the challenge inputs based on the challenger Ch's challenge bit. We construct the anonymity functions for different anonymity notions in Section 2.3.

About the Additive and Multiplicative Bound In Definition 2.2.1, we bound the difference in probabilities between $\langle \mathrm{A}(1^\eta)|\textsc{Ch}(\Pi, \alpha, \gamma, 0)\rangle$ and $\langle \mathrm{A}(1^\eta)|\textsc{Ch}(\Pi, \alpha, \gamma, 1)\rangle$ by a multiplicative bound e^ε and an additive bound δ. The multiplicative bound indicates the distinguishability of both random variables under ideal conditions, i.e. quantifies the leakage inherent to the analyzed protocol Π, even against a passively observing adversary. The additive bound δ gives us the probability with which the adversary can (with certainty higher than ε) distinguish both $\textsc{Ch}(\Pi, \alpha, \gamma, 0)$ and $\textsc{Ch}(\Pi, \alpha, \gamma, 1)$. For anonymous communication networks this can be intuitively interpreted as the likelihood of a *distinguishing event* which allows the adversary to break anonymity and identify the hiding party.

If the adversary has an advantage δ in distinguishing between two scenarios, e.g., two possible recipients of a message, this might stem from a chance of breaking the cryptography involved in the protocol, i.e., by guessing a decryption key, or a chance to entirely compromise relevant protocol parties. Generally, such an advantage can stem from a certain probability of finding hard evidence or proof for the scenario. In contrast, a (purely) multiplicative factor describes that the adversary has some advantage in guessing the scenario, but is provably not able to find hard evidence or even proof.

Multiple Challenges As many differential-privacy style guarantees, *IND-CDP* benefits from a straight-forward generalization on the number of challenges. Allowing more than one challenge does not lead to conceptually new attacks, but simply increases the values of ε and δ depending on the number of challenges:

Informal Theorem 2.2.1 (Single-Challenge Reducibility)**.** *For every protocol* Π*, every anonymity function* α*, and every* $\gamma \in \mathbb{N}$*. Whenever* Π *is* $(\alpha, 1, \varepsilon, \delta)$*-IND-CDP, with* $\varepsilon \geq 0$ *and* $0 \leq \delta \leq 1$*, then* Π *is* $(\alpha, \gamma, \gamma \cdot \varepsilon, \gamma \cdot e^{\gamma\varepsilon} \cdot \delta)$*-IND-CDP.*

Note that we did not specify the adversary in this Theorem, as its validity cannot be tied to an individual adversary, but requires a class of adversaries. Consequently, we do not discuss the generalization here, but refer to Section 2.4.1, where we define adversary classes for which the Theorem holds. Such a result greatly amplifies the utility of our analyses: a rather simple analysis yields guarantees for an arbitrary number of challenges.

2.3 Anonymity Notions

In this section we present a variety of example anonymity notions for AnoA. These notions are represented as *anonymity functions* that describe the challenges in the game-based definition between our challenger (see Figure 2.1) and the adversary.

For all anonymity notions, the *challenge state* `state` defines whether a challenge has not been started yet (in which case `state` = `FRESH`) and otherwise carries all relevant information for ensuring consistency over all messages belonging to the same challenge.

```
$\alpha_{\mathsf{SA}}(\mathsf{state}, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$ :
  if $\mathsf{state} = \mathtt{FRESH}$ then
    $\mathsf{state} := (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0)$
  else if $\mathsf{state} \neq (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0)$ then
    output $\perp$
  end if
  output $(\mathsf{state}, \mathsf{S}_b, \mathsf{R}_0)$

$\alpha_{\mathsf{RA}}(\mathsf{state}, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$ :
  if $\mathsf{state} = \mathtt{FRESH}$ then
    $\mathsf{state} := (\mathsf{S}_0, \mathsf{R}_0, \mathsf{R}_1)$
  else if $\mathsf{state} \neq (\mathsf{S}_0, \mathsf{R}_0, \mathsf{R}_1)$ then
    output $\perp$
  end if
  output $(\mathsf{state}, \mathsf{S}_0, \mathsf{R}_b)$

$\alpha_{\mathsf{REL}}(\mathsf{state}, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$ :
  if $\mathsf{state} = \mathtt{FRESH}$ then
    $a \leftarrow^u \{0, 1\}$
    $\mathsf{state} := (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, a)$
  else if $\mathsf{state} \neq (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, a)$ for some $a \in \{0, 1\}$ then
    output $\perp$
  end if
  if $b = 0$ then
    output $(\mathsf{state}, \mathsf{S}_a, \mathsf{R}_a)$
  else
    output $(\mathsf{state}, \mathsf{S}_a, \mathsf{R}_{1-a})$
  end if
```

Figure 2.2: The anonymity functions for AnoA.

2.3.1 Sender Anonymity

Sender anonymity characterizes the anonymity of senders against a (possibly even malicious) recipient. In contrast to other notions from the literature, we define sender anonymity as the inability of an observer to decide which of two *self-chosen* senders is sending messages to the recipient.

We formalize our notion of sender anonymity with the definition of an anonymity function α_{SA} as depicted in Figure 2.2. Basically, α_{SA} selects one of two possible challenge senders while returning the first given recipient.

Definition 2.3.1 (Sender anonymity). *A protocol* Π *provides* $(\gamma, \varepsilon, \delta)$-sender anonymity *against* A*, if it is* $(\alpha_{\mathsf{SA}}, \gamma, \varepsilon, \delta)$*-IND-CDP against* A *for* α_{SA} *as defined in Figure 2.2.*

Example 2.3.1 (Sender anonymity). *The adversary* A *decides that it wants to use the senders Alice and Bob in the sender anonymity game. It sends a challenge* $\texttt{Challenge} =$ $(\text{Alice}, \text{Bob}, \text{evilserver.com}, \text{evilserver.com}, m, 1)$ *for sending some message* m *of* A*'s choice to a (probably corrupted) recipient evilserver.com; note that there is only one relevant recipient for a sender anonymity challenge. The anonymity function* α_{SA} *now outputs (depending on the challenge bit) one of the two senders and the recipient.*

2.3.2 Recipient Anonymity

Recipient anonymity characterizes that the recipient of a communication remains anonymous, even to observers that have knowledge about the sender in question. Our notion of recipient anonymity is defined analogously to sender anonymity: the anonymity function selects one of two possible recipients for a message.

Definition 2.3.2 (Recipient anonymity). *A protocol* Π *provides* $(\gamma, \varepsilon, \delta)$-recipient anonymity *against* A*, if it is* $(\alpha_{\mathsf{RA}}, \gamma, \varepsilon, \delta)$*-IND-CDP against* A *for* α_{RA} *as defined in Figure 2.2.*

Remark 2.3.1 (Messages and Website Fingerprinting). *There are known practical attacks against recipient anonymity that work by learning the* fingerprint *of a website, i.e., the shape of messages sent by the respective web-server. In such a website fingerprinting attack, the attacker checks traffic features (e.g., volume and direction changes) of the response, which in many cases is unique as recent studies show [92, 25]. Such a trivial attack is possible against any low-latency anonymous communication protocol and should hence be excluded for quantifying the anonymity guarantees of a low-latency anonymous communication protocol. Following our intuition of anonymity from Section 1.2.1 that an ideal AC protocol should provide an anonymous channel, not necessarily hide the identity of parties that leak information about themselves via the content of messages they send, we exclude such attacks by explicitly requiring that the content of messages does not depend on the challenge bit.*

One can account for possibly different message requirements for different recipients, by introducing two challenge messages, similar to senders and recipients. The anonymity function then selects the message appropriate for the selected recipient. To still counter attacks based on the pattern of those messages, the challenger checks whether the messages have the same length.

2.3.3 Relationship Anonymity

A protocol satisfies *relationship anonymity*, if for any communication, the adversary cannot determine sender and recipient of this communication at the same time [93]. We model this property by letting the anonymity function α_{REL} choose one of two possible senders and one of two possible recipients for a given challenge. We then group the four possible combinations in two sets (one defined by the adversary and one modified by the anonymity function) and let the adversary guess from which set the challenger sampled the sender and the recipient of the communication.

More formally, the anonymity function α_{REL} on input $(\mathsf{state}, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$, as defined in Figure 2.2, chooses either a sender S_i and a recipient R_j with the same index $i = j$ (for $b = 0$) or a sender S_i and a recipient R_j with different indexes $i \neq j$ (for $b = 1$).

Definition 2.3.3 (Relationship anonymity)**.** *A protocol* Π *provides* $(\gamma, \varepsilon, \delta)$-relationship anonymity *against* A*, if it is* $(\alpha_{\mathsf{REL}}, \gamma, \varepsilon, \delta)$*-IND-CDP against* A *for* α_{REL} *as defined in Figure 2.2.*

Example 2.3.2 (Relationship anonymity)**.** *The adversary* A *decides that it wants to use the senders Alice and Charlie and the recipients Bob and Dave in the relationship anonymity game. It wins the game if it can distinguish between the scenario "0" where Alice communicates to Bob or Charlie communicates to Dave and the scenario "1" where Alice communicates to Dave or Charlie communicates to Bob. Only one of those four possible instructions will be fed to the protocol.*

The adversary A *sends a challenge message* `Challenge` $= (\text{Alice}, \text{Charlie}, \text{Bob}, \text{Dave}, m, \Psi)$ *to* Ch. *The anonymity function* α_{REL} *then randomly selects a sender* $a \in \{\text{Alice}, \text{Charlie}\}$*, say Alice. Then, depending on the challenge bit b, the function outputs either the pair* $(\text{Alice}, \text{Bob})$ *as specified by* A *(for* $b = 0$*), or* $(\text{Alice}, \text{Dave})$ *as a modified instruction (for* $b = 1$*).*

2.4 Adversary Classes

AnoA assumes a strong attacker that can choose the users' inputs and, if the protocol permits, freely compromise protocol parties and connections between them. For many protocols the resulting anonymity guarantees are quantitative and inherently depend on the choices of the adversary, most prominently on its compromisation strategy. By defining restrictions for the adversary's capabilities, we can analyze a multitude of realistic scenarios and give meaningful worst-case guarantees for all adversaries with these restrictions.

Technically, we reason about sets of adversaries that do not exploit their full capabilities. We call such sets *adversary classes* if they can be constructively described in terms of a `PPT` wrapper machine that internally runs arbitrary `PPT` machines. In this section, we give examples for adversary classes and identify sanity conditions for adversary classes. We show that all adversary classes that satisfy these sanity conditions are *single-challenge reducible* (SCR). For a SCR adversary class it suffices to show anonymity bounds for a single challenge to immediately derive bounds for more challenges, which significantly simplifies anonymity analyses. Finally, to ease the usage of AnoA in subsequent analyses, we construct a general template for adversary classes (*Plug'n'Play* adversary classes) and show that every instantiation of the template is SCR.

2.4.1 Defining Adversary Classes

An *adversary class* is a set of `PPT` machines that is constructively defined in terms of a `PPT` wrapper machine $\mathcal{A}(\cdot)$. This wrapper machine internally simulates the adversary in a black-box fashion, which we denote with $\mathcal{A}(\mathrm{A})$ for any specific adversary A. The machine $\mathcal{A}(\mathrm{A})$ restricts A in the following three dimensions: (a) in its ability to send messages to the protocol (i.e., for compromising parties or injecting new messages), (b) in its possible challenge messages, and thus, in its knowledge about the world and (c) in its possible observations of messages from the protocol, and thus, in its ability to gain insights about leakage. Moreover it may communicate with the adversary. We simply write $\mathcal{A}$ for the set of all `PPT` adversaries described by the wrapper machine: $\mathcal{A} = \{\mathcal{A}(\mathrm{A}) | \mathrm{A} \text{ is a } \texttt{PPT} \text{ machine}\}$.

Definition 2.4.1 (Anonymity Guarantees against Adversary Classes)**.** *For every anonymity notion α, we say that a protocol is $(\alpha, \gamma, \epsilon, \delta)$-IND-CDP against an adversary class $\mathcal{A}$, if is $(\alpha, \gamma, \epsilon, \delta)$-IND-CDP against every $\mathrm{A} \in \mathcal{A}$.*

The following example highlights a scenario in which restricting the adversary (not only in its compromisation) is important for an analysis.

Example 2.4.1 (Tor with entry guards)**.** *Consider Tor with so-called* entry guards. *Every user selects only one entry node (his guard) that serves as the entry node of every circuit. A new guard is chosen only after several months. As a compromised entry node is fatal for the security guarantees that Tor can provide, the concept of entry guards helps in reducing the risk of choosing a malicious node. However, if such an entry guard is compromised the impact is more severe since an entry guard is used for a large number of circuits.*

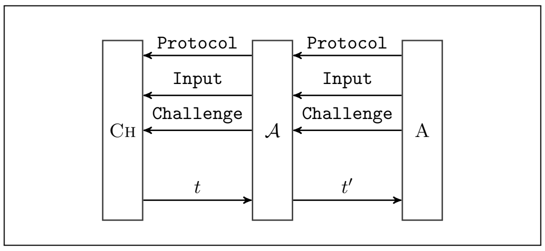
The adversary can choose its targets adaptively and thus perform the following attack. Initially it corrupts some nodes that can in principle be entry guards and then sends (polynomially many) messages $\texttt{Input} = (\mathsf{S}, \mathsf{R}, m)$ *for different users* S*, until one of them, say Alice, chooses a compromised node as its entry guard. Then* A *proceeds by choosing Alice and some other user in a challenge. As Alice will use the compromised entry guard again, the adversary trivially wins the sender anonymity game.*

This extremely successful attack is not unrealistic: It models the fact that if there are compromised entry guards, then certainly some of the millions of Tor users will *use them and consequently those unlucky users* will *be deanonymized by the adversary. However, this might not be the attack scenario that we are interested in. Thus, we define an adversary class that makes sure that the adversary cannot choose its targets. Whenever the adversary starts a new challenge for sender anonymity, the adversary class draws two users at random and places them into the challenge messages of this challenge.*

Adversary classes are a versatile tool to restrict the adversary's capabilities. Although adversary classes are critical to the results of an analysis, slight mistakes in defining an adversary class can lead to unintuitive results. Recall that we strive for *single-challenge reducibility*, i.e., we strive to show that *IND-CDP* against adversaries that only use one challenge immediately implies *IND-CDP* against adversaries that use more than one challenge. The quantitative guarantees naturally depend on the number of challenges.

The set of all adversaries (not restricted by an adversary class) satisfies single-challenge reducibility and thus for multiple challenges yields a degradation of anonymity similar to differential privacy: Both ε and δ increase linearly in the number of challenges in the same way in which a larger distance between databases linearly increases the guarantees for differential privacy. The intuition behind this generalization is similar to comparable guarantees for differential privacy: As long as the protocol is oblivious to whether it was instructed by a challenge or an input message, any additional challenge can increase the advantage of an adversary (linearly), but cannot allow the adversary to perform attacks that were impossible without this additional challenge. Intuitively, if there was a strong attack against the protocol that required $\gamma \geq 2$ challenges, then the adversary could simulate this attack although it only initiates one challenge: The adversary randomly samples a challenge bit b_A and then simulates the first $\gamma - 1$ challenge messages by computing the anonymity function on them, using b_A as the challenge bit, and sending the resulting messages as an input messages. Finally, it sends the last challenge as in the original attack. If the guess of the challenge bit was correct, the protocol receives exactly the same messages in this 1-challenge-attack as it receives in the original γ-challenge-attack. Consequently, the attack can be performed. Adversary classes, however, can treat challenge messages in a special way such that anonymity guarantees established for a single challenge tag cannot be generalized to several challenge tags. Hence, allowing the adversary to initiate more challenges can allow for new attacks instead of linearly increasing the advantage. The next example illustrates this problem.

Example 2.4.2 (Non-single-challenge reducible adversary class)**.** *Consider the following adversary class $\mathcal{A}_{\mathtt{nc}}$. The adversary class relays input messages to the challenger and replaces the messages of the first challenge by (say) random strings or error symbols. The adversary class simply relays all messages to the challenger except for the ones belonging to the first challenge.*



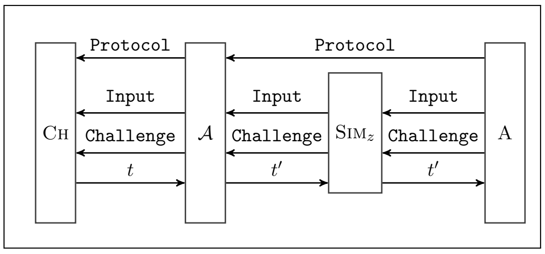


Figure 2.3: The two games relevant for our proof: the real game `AC-Real` (left), where A directly communicates with $\mathcal{A}$ and the simulator game `AC-Sim`[SIM$_z$] (right), where A communicates with SIM$_z$ that in turn communicates with $\mathcal{A}$. In this figure, `Protocol` specifies messages that are sent directly to the protocol and not to the challenger.

Every protocol, even if completely insecure, will be IND-CDP secure for one challenge for the class $\mathcal{A}_{\texttt{nc}}$ (as the adversary cannot gain any information about the bit b of the challenger), but it might not necessarily be secure for more than one challenge.

Requirements for Single-Challenge Reducibility (SCR)

Showing that an adversary class (in general) satisfies SCR is not as simple as it might appear on first glance. Technically, we show that even maliciously described protocols cannot violate SCR. To this end, we show that SCR can be achieved if the adversary can—despite potential modifications of the adversary class—cause the same protocol behavior with input messages as with challenge messages. More precisely, we require that the entire behavior of the adversary class, the challenger and the protocol on challenges can be simulated by an adversary that guesses the challenge bit correctly and sends inputs instead. To this end, we introduce a machine SIM, called the challenge-simulator that sits between the adversary and the adversary class and that simulates challenge messages (sent by the adversary) with input messages.

Technically, we compare the normal scenario `AC-Real` with the challenger, the adversary class, and the adversary to the scenario `AC-Sim` with the challenger, the adversary class, the challenge simulator, and the adversary. We consider not only the case where all challenge messages are simulated with input messages but also all case in which only parts of the challenge messages are simulated. Hence, we require a family of challenge simulators, indexed by a sequence z of n pairs (z_i, b_i). If $z_i =$ `don't simulate`, the challenge simulator SIM does not simulate the challenge messages of the ith challenge tag. If $z_i =$ `simulate`, the challenge simulator simulates the challenge messages of the ith challenge tag as if the challenge recommendation bit b_i was the challenge bit. Construction 1 describes both scenarios in a more technical manner.

Construction 1. *Consider the following two scenarios (as depicted in Fig. 2.3):*

- `AC-Real`(Π, α, γ, b)*:* A *communicates with* $\mathcal{A}$ *and* $\mathcal{A}$ *communicates with* CH$(\Pi, \alpha,$

$\gamma, b)$. The bit of the challenger is b and the adversary may send challenge tags in $\{1, \dots, n\}$.

- $\texttt{AC-Sim}[\textsc{Sim}_z](\Pi, \alpha, \gamma, b)$*:* A *communicates with* $\textsc{Sim}_z$ *that in turn communicates with* $\mathcal{A}$ *and* $\mathcal{A}$ *communicates with* $\textsc{Ch}(\Pi, \alpha, \gamma, b)$*.The bit of the challenger is b and the adversary may send challenge tags in $\{1, \dots, \gamma\}$.*

In the proof, we use the simulator in an intricate hybrid argument. For that argument, we additionally need that a simulation using the wrong bit for some challenges can still simulate other challenges for the correct bit. For a given challenge bit b and for any pair of simulator indices z, z', the scenarios $\texttt{AC-Sim}[\textsc{Sim}_z](\Pi, \alpha, \gamma, b)$ and $\texttt{AC-Sim}[\textsc{Sim}_{z'}](\Pi, \alpha, \gamma, b)$ are indistinguishable if: whenever z differs from z' in the simulation bit ($z_i \neq z'_i$) for some position i and $z_i = \texttt{simulate}$, the corresponding challenge recommendation bit $b_i = b$ is consistent with the challenge bit.

Definition 2.4.2 (Consistent simulator index)**.** *A* simulator index *(for γ challenges) is a bitstring $z = [(z_1, b_1), \dots, (z_\gamma, b_\gamma)] \in \{0,1\}^{2\gamma}$. A pair of simulator indexes $z, z' \in \{0,1\}^{2\gamma}$ (for γ challenges) is consistent w.r.t. b if $\forall i \in \{1, \dots, \gamma\}$ s.t. $z_i \neq z'_i$ we have*

$$(z_i = \texttt{simulate} \Rightarrow b_i = b) \wedge (z'_i = \texttt{simulate} \Rightarrow b'_i = b).$$

Additionally, we require that the adversary class must not initiate challenges on its own, though it can drop challenges (*reliability*). Moreover, the adversary class is independent from the actual challenge tags per se, i.e., it does not interpret the challenge tags in a semantic way (*alpha-renaming*).

Definition 2.4.3 (Condition for adversary class)**.** *An adversary class $\mathcal{A}$ is* single-challenge reducible *(SCR) for an anonymity function α, if the following conditions hold:*

1. ***Reliability:*** *$\mathcal{A}(\mathrm{A})$ never sends a message $\texttt{Challenge} = (\dots, \Psi)$ to the challenger before receiving a message $\texttt{Challenge} = (\dots, \Psi)$ from A with the same challenge tag Ψ.*

2. ***Alpha-renaming:*** *$\mathcal{A}$ does not behave differently depending on the challenge tags Ψ that are sent by A except for using it in its own messages $\texttt{Challenge} = (\dots, \Psi)$ to the challenger. More formally, for every permutation π on $\mathbb{N}$, we define $\mathcal{A}^\pi$ as the machine that replaces all challenge tags Ψ within messages reaching $\mathcal{A}$ by $\pi(\Psi)$ and all messages sent from $\mathcal{A}$ containing a challenge tag Ψ by $\pi^{-1}(\Psi)$ and that otherwise simulates $\mathcal{A}$. For all such machines, $\mathcal{A}^\pi$ behaves exactly as $\mathcal{A}$, i.e., for each input, its outputs have the same distribution.*

3. ***Simulatability:*** *For every $n \in \mathbb{N}$ and every list $z = [(z_1, b_1), \dots, (z_n, b_n)] \in \{0,1\}^{2n}$ there exists a machine $\textsc{Sim}_z$ such that:*

(a) *For every $i \in \{1, \ldots, n\}$, the following holds: if $z_i =$ `simulate`, then* $\textsc{Sim}_z$ *never sends a message* `Challenge` $= (\ldots, i)$ *to* $\mathcal{A}$.

(b) *The games* `AC-Real`(Π, α, γ, b) *and* `AC-Sim`$[\textsc{Sim}_{z_{\texttt{real}}}](\Pi, \alpha, \gamma, b)$*, as defined below are computationally indistinguishable, where* $z_{\texttt{real}} = [($`don't simulate`$, \_), \ldots, ($`don't simulate`$, \_)] \in \{0,1\}^{2n}$ *for* A.

(c) *for all consistent simulator indices $z, z' \in \{0,1\}^{2n}$ (see Definition 2.4.2), the games* `AC-Sim`$[\textsc{Sim}_z](\Pi, \alpha, \gamma, b)$ *and* `AC-Sim`$[\textsc{Sim}_{z'}](\Pi, \alpha, \gamma, b)$ *are computationally indistinguishable for* A.

Theorem 2.4.1. *Let Π be a protocol, α be an anonymity function, $\gamma \in \mathbb{N}$ be a number of challenges, and $\mathcal{A}$ be an adversary class that is SCR for α, as in Definition 2.4.3. Whenever Π is $(\alpha, 1, \varepsilon, \delta)$-IND-CDP for $\mathcal{A}(\mathrm{A})$, with $\varepsilon \geq 0$ and $0 \leq \delta \leq 1$, then Π is $(\alpha, \gamma, \gamma \cdot \varepsilon, \gamma \cdot e^{\gamma\varepsilon} \cdot \delta)$-IND-CDP for $\mathcal{A}(\mathrm{A})$.*

Proof. We show the composability theorem by induction over γ. We assume the theorem holds for i and compute the anonymity loss between the games `AC-Real`$(\Pi, \alpha, i+1, 0)$, where we have $b = 0$ and `AC-Real`$(\Pi, \alpha, i+1, 1)$, where we have $b = 1$ via a transition of indistinguishable, or differentially private games.

Proof Structure: We start with `AC-Real`$(\Pi, \alpha, i+1, 0)$ and introduce a simulator that simulates one of the challenges for the correct bit $b = 0$. We apply the induction hypothesis for the remaining i challenges (this introduces an anonymity loss of $(i \cdot \varepsilon, i \cdot e^{i \cdot \varepsilon} \cdot \delta)$). The simulator still simulates one challenge for $b = 0$, but the bit of the challenger is now $b = 1$. We then simulate all remaining n challenges for $b = 1$ and thus introduce a game in which all challenges are simulated. As the bit of the challenger is never used in the game, we can switch it back to $b = 0$ again and remove the simulation of the first challenge. We can apply the induction hypothesis again (we loose (ε, δ)) and switch the bit of the challenger to $b = 1$ again. In this game, we have one real challenge (for $b = 1$) and n simulated challenges (also for $b = 1$). Finally, we remove the simulator again and yield `AC-Real`$(\Pi, \alpha, i+1, 1)$.

Assume that Π is $(i, i \cdot \varepsilon, e^{i\varepsilon} \cdot i \cdot \delta)$-*IND-CDP*. We show that Π is also $(i+1, (i+1) \cdot \varepsilon, e^{(i+1) \cdot \varepsilon}(i+1) \cdot \delta)$-*IND-CDP*. Let A be an adversary that sends at most $i+1$ challenges. We construct several games:

- **Game:** G_0 is the normal game `AC-Real`$(\Pi, \alpha, i+1, 0)$ with up to $i+1$ challenges where $b = 0$.

- **Game:** G_1 is an intermediate game `AC-Sim`$[\textsc{Sim}_{z_{\texttt{real}}}](\Pi, \alpha, i+1, 0)$. Here every message from A to $\mathcal{A}(\mathrm{A})$ (and otherwise) goes through the simulator $\textsc{Sim}_{z_{\texttt{real}}}$. However, by definition of $z_{\texttt{real}}$, the simulator does not need to simulate any challenge.

  Claim: G_0 and G_1 are computationally indistinguishable.

  Proof: By item 3b Definition 2.4.3 the simulator $\textsc{Sim}_{z_{\texttt{real}}}$ exists and the games are indistinguishable.

- **Game:** G_2 is an intermediate (hybrid) game $\texttt{AC-Sim}[\textsc{Sim}_z](\Pi, \alpha, i+1, 0)$ with $b = 0$ and fixed `Input` messages instead of the challenge with tag $i+1$ (so there are at most i challenges left). This is done by using the simulator $\textsc{Sim}_z$ for $z = [(\texttt{don't simulate}, \_), \ldots, (\texttt{don't simulate}, \_), (\texttt{simulate}, 0)] \in \{0,1\}\, i+1$, i.e., the simulator simulates the $i+1$st challenge for $b = 0$.

  Claim: G_1 and G_2 are computationally indistinguishable.

  Proof: By item 3 of Definition 2.4.3, we know that the simulator $\textsc{Sim}_z$ exists. Since the simulator $\textsc{Sim}_z$ from G_2 uses the correct bit $b_{i+1} = 0$ for the simulated challenge, Item 3c of Definition 2.4.3 implies that the games are indistinguishable.

- **Game:** G_3 is the intermediate (hybrid) game $\texttt{AC-Sim}[\textsc{Sim}_z](\Pi, \alpha, i+1, 1)$ where the simulator stays $\textsc{Sim}_z$ but the challenger changes to $b = 1$.

  Claim: G_2 and G_3 are $(i\varepsilon, e^{i\varepsilon} i\delta)$-indistinguishable.

  Proof: The adversary $\textsc{Sim}_z(\mathrm{A})$ makes at most i queries with challenge tags in $\{1, \ldots, i\}$. From the reliability property of the adversary class (item 1 of Definition 2.4.3) we know that thus $\mathcal{A}(\textsc{Sim}_z(\mathrm{A}))$ uses at most i challenge tags in $\{1, \ldots, i\}$. The claim immediately follows from the induction hypothesis: Π is $(i, i \cdot \varepsilon, i \cdot \delta)$-*IND-CDP*.

- **Game:** G_4 is a game $\texttt{AC-Sim}[\textsc{Sim}_{z'}](\Pi, \alpha, i+1, 1)$ where the simulator $\textsc{Sim}_{z'}$ with $z' = [(\texttt{simulate}, 1), \ldots, (\texttt{simulate}, 1), (\texttt{simulate}, 0)]$ simulates all challenges from A. For the challenge tags 1 to i, $\textsc{Sim}_{z'}$ simulates the challenges for $b_1 = \ldots = b_i = 1$, whereas for the tag $i+1$ it still simulates it for $b_{i+1} = 0$. The challenger uses $b = 1$.

  Claim: G_3 and G_4 are computationally indistinguishable.

  Proof: Since the simulator $\textsc{Sim}_{z'}$ from G_4 uses the correct bit $b_1 = \ldots = b_i = 1$ for the challenges that are not simulated in $\textsc{Sim}_z$, Item 3c of Definition 2.4.3 implies that the games are indistinguishable.

- **Game:** G_5 is the game $\texttt{AC-Sim}[\textsc{Sim}_{z'}](\Pi, \alpha, i+1, 0)$ where we use the same simulator $\textsc{Sim}_{z'}$ as in G_4, but the challenge bit is set to $b = 0$ again.

  Claim: G_4 and G_5 are computationally indistinguishable.

  Proof: Since there are no challenge messages (everything is simulated, as by item 3a $\textsc{Sim}_{z'}$ does not send any messages $\texttt{Challenge} = (\ldots, \Psi)$), changing the bit b of the challenger does not have any effect. Hence, the games are indistinguishable.

- **Game:** G_6 is the game $\texttt{AC-Sim}[\textsc{Sim}_{z''}](\Pi, \alpha, i+1, 0s)$ where we use the simulator $\textsc{Sim}_{z''}$ with $z'' = [(\texttt{simulate}, 1), \ldots, (\texttt{simulate}, 1), (\texttt{don't simulate}, \_)]$. In other words, we do not simulate the challenge for $i+1$ with $b_{i+1} = 0$, but we use the challenger again (also with $b = 0$). Still, the challenges from 1 to i are simulated with $b_1 = \ldots = b_i = 1$.

Claim: G_5 and G_6 are computationally indistinguishable.

Proof: Since the simulator $\textsc{Sim}_{z'}$ from G_5 uses the correct bit $b_{i+1} = 0$ for the simulated challenge (which the simulator $\textsc{Sim}_{z''}$ does not simulate), Item 3c of Definition 2.4.3 implies that the games are indistinguishable.

- **Game:** G_7 is $\texttt{AC-Sim}[\textsc{Translator}(\textsc{Sim}_{z''})](\Pi, \alpha, i+1, 0)$ where we build around the simulator $\textsc{Sim}_{z''}$ an interface $\textsc{Translator}(\cdot)$ that translates the challenge tag from $i+1$ to 1 and vice versa in all messages $\texttt{Challenge} = (\ldots, \Psi)$ from $\textsc{Sim}_{z''}$ to $\mathcal{A}(\cdot)$ and in all messages $\texttt{Observations} = (t, \Psi)$ from $\mathcal{A}(\cdot)$ to $\textsc{Sim}_{z''}$.

  Claim: G_6 and G_7 are information theoretically indistinguishable.

  Proof: Item 2 of Definition 2.4.3 requires that the renaming of challenge tags does not influence the behavior of $\mathcal{A}(\cdot)$. It also does not influence the behavior of the challenger (by definition) or the protocol (that never sees challenge tags). Thus, the games are indistinguishable.

- **Game:** G_8 is the game $\texttt{AC-Sim}[\textsc{Translator}(\textsc{Sim}_{z''})](\Pi, \alpha, i+1, 1)$ where the simulator is defined as in G_7 but the challenge bit is set to $b = 1$.

  Claim: G_7 and G_8 are (ε, δ) indistinguishable.

  Proof: By assumption of the theorem, the protocol Π is $(1, \varepsilon, \delta)$-*IND-CDP* for $\mathcal{A}(\mathrm{A})$. Moreover, by definition of z'' and by item 3a, the translated adversary $\textsc{Translator}(\textsc{Sim}_{z''}(\mathrm{A}))$ only uses at most one challenge tag, namely the tag 1. From the reliability property of the adversary class (item 1 of Definition 2.4.3) we know that thus $\mathcal{A}(\textsc{Translator}(\textsc{Sim}_{z''}(\mathrm{A})))$ uses only the challenge tag 1. Thus, G_7 and G_8 are (ε, δ) indistinguishable.

- **Game:** G_9 is $\texttt{AC-Sim}[\textsc{Sim}_{z''}](\Pi, \alpha, i+1, 1)$, defined like G_8, except that we remove the translation interface again.

  Claim: G_8 and G_9 are information theoretically indistinguishable.

  Proof: As before, Item 2 of Definition 2.4.3 requires that the renaming of challenge tags does not influence the behavior of $\mathcal{A}(\mathrm{A})$. It also does not influence the behavior of the challenger (by definition) or the protocol (that never sees challenge tags). Thus, the games are indistinguishable.

- **Game:** G_{10} is the target game $\texttt{AC-Real}(\Pi, \alpha, i+1, 1)$ without modifications, where the challenge bit is $b = 1$.

  Claim: G_9 and G_{10} are computationally indistinguishable.

  Proof: Since $\textsc{Sim}_{z''}$ uses the correct bit $b_1 = \ldots = b_i = 1$ for all simulations, we can replace it with $\textsc{Sim}_{z_{\texttt{real}}}$, that, in turn, is indistinguishable from $\texttt{AC-Real}(\Pi, \alpha, i+1, 1)$.

We slightly abuse notation in writing $\texttt{Pr}\left[0 = \mathrm{A}(G_0)\right]$ for

$$\texttt{Pr}\left[0 = \langle \mathcal{A}(\mathrm{A}) | \textsc{Ch}(\Pi, \alpha, \gamma, 0) \rangle\right],$$

$\texttt{Pr}\left[0 = \mathrm{A}(G_1)\right]$ for

$$\texttt{Pr}\left[0 = \langle \mathcal{A}(\textsc{Sim}_z(b, \mathrm{A})) | \textsc{Ch}(\Pi, \alpha, \gamma, 0) \rangle\right], \text{ etc..}$$

We can then show the Theorem via the following inequations, where we denote the negligible functions induced by the (computational) indistinguishability properties by μ. We neglect the (implicit) security parameter 1^η of the adversary, the simulator and the challenger and stress that the last inequality holds for a sufficiently large security parameter.

$$\begin{aligned}
&\texttt{Pr}\left[0 = \mathrm{A}(G_0)\right] \\
&\leq \texttt{Pr}\left[0 = \mathrm{A}(G_1)\right] + \mu_1 \\
&\leq \texttt{Pr}\left[0 = \mathrm{A}(G_2)\right] + \mu_2 + \mu_1 \\
&\leq e^{i\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_3)\right] + e^{i\varepsilon} i\delta + \mu_2 + \mu_1 \\
&\leq e^{i\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_4)\right] + e^{i\varepsilon}(\mu_3 + i\delta) + \mu_2 + \mu_1 \\
&\leq e^{i\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_5)\right] + e^{i\varepsilon}(\mu_4 + \mu_3 + i\delta) + \mu_2 + \mu_1 \\
&\leq e^{i\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_6)\right] + e^{i\varepsilon}(\mu_5 + \mu_4 + \mu_3 + i\delta) + \mu_2 + \mu_1 \\
&= e^{i\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_7)\right] + e^{i\varepsilon}(\mu_5 + \mu_4 + \mu_3 + i\delta) + \mu_2 + \mu_1 \\
&\leq e^{i\varepsilon}(e^{\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_8)\right] + \delta) + e^{i\varepsilon}(\mu_5 + \mu_4 + \mu_3 + i\delta) + \mu_2 + \mu_1 \\
&= e^{(i+1)\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_8)\right] + e^{i\varepsilon}(\mu_5 + \mu_4 + \mu_3 + (i+1)\delta) + \mu_2 + \mu_1 \\
&= e^{(i+1)\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_9)\right] + e^{i\varepsilon}(\mu_5 + \mu_4 + \mu_3 + (i+1)\delta) + \mu_2 + \mu_1 \\
&\leq e^{(i+1)\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_{10})\right] + e^{(i+1)\varepsilon}\mu_6 + e^{i\varepsilon}(\mu_5 + \mu_4 + \mu_3 + (i+1)\delta) + \mu_2 + \mu_1 \\
&\leq e^{(i+1)\varepsilon}\texttt{Pr}\left[0 = \mathrm{A}(G_{10})\right] + e^{(i+1)\varepsilon}(i+1)\delta
\end{aligned}$$

This concludes the proof. □

2.4.2 Plug'n'Play Adversary Classes

Proving single-challenge reducibility can be tedious and lengthy. Therefore, we develop a template for adversary classes, called the *Plug'n'Play* (PnP) template and show that every adversary class that can be expressed in this template is single-challenge reducible.

The Plug'n'Play template forwards all input messages from the adversary and starts for every challenge tag a new submachine, called a *challenge machine*, with a fresh memory. In particular, none of the challenge machines shares memory with any other part of the adversary class. These challenge machines can modify challenges and create fresh input messages, as well as communicate with the adversary. Additionally, the template includes a provision for blocking messages in both directions via a so-called *filter* machine, that generalizes limitations on compromisation, injection, and observation. The setup-machine S allows to specify initial adversarial messages (before any challenge or input message). Given such a challenge machine M, a filter F, and a setup machine S we construct a PnP adversary class $\textsc{PnP}_{M,F,S}$ as follows.

Definition 2.4.4 (PnP Adversary Class). *For three* PPT *machines M, the* challenge machine*, F, the* filter*, and S, the* setup*, we define the* Plug'n'Play adversary class $\textsc{PnP}_{M,F,S}$ *as follows.*

1. *Initially send all messages* $\texttt{setup}(m)$ *from* A *to S (and block all other messages from* A*) and send all messages from S to* A *and to* Π *until S stops. Block all messages from S to the challenger that are of the type* $\texttt{Input}(\ldots)$ *or* $\texttt{Challenge}(\ldots)$*: S may only receive messages from* A *and* Π *and send messages to* A *and* Π*; S can never send a message* $\texttt{Input}$ *or* $\texttt{Challenge}$*.*

2. *Whenever* A *sends a message* $\texttt{Input} = (\mathsf{S}, \mathsf{R}, m, (\mathrm{A}, \mathsf{z}))$ *for any nonce* z*, relay this message as* $\texttt{Input} = (\mathsf{S}, \mathsf{R}, m, \mathsf{z})$ *to the challenger.*

3. *Whenever* A *sends a message* $\texttt{Input} = (\mathsf{S}, \mathsf{R}, m, (\textsc{Ch}, \Psi))$ *or a message* $\texttt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, m, (\textsc{Ch}, \Psi))$*, run M_Ψ on the respective message* $\texttt{Input}/\texttt{Challenge}$*, as below.*

4. *Whenever* A *sends any message m to the protocol* Π *directly, run F on m until it outputs a message m' or an error symbol $\perp$. If it outputs a message m', send m' to* Π*.*

5. *Whenever* Π *sends a message t, first simulate F on t until it outputs a message t', or an error symbol $\perp$. If it returned t', relay it to* A*.*

6. *Whenever* $\textsc{Ch}$ *sends a message* $\texttt{Observations}(m, \mathsf{ID})$*, relay this message to* A*.*

Running M_Ψ on a message m

- *If M_Ψ was not initialized so far (i.e., if a message with* $\mathsf{ID} = (\textsc{Ch}, \Psi)$ *is received for the first time), initialize a new instance of the machine M as M_Ψ.*
- *Then / otherwise simulate M_Ψ on m (without its message* ID*, i.e., replace* $(\textsc{Ch}, \Psi)$ *by $\perp$), until M_Ψ outputs a message m' and relay m', depending on its structure: to* A *(if $m' =$* $\texttt{Observations}(t, \mathsf{ID}')$*), or to* $\textsc{Ch}$ *(otherwise). In both cases, replace the message identifier* ID' *by* $(\textsc{Ch}, \Psi)$*.*

On the efficiency of a PnP adversary We only consider efficient probabilistic Turing machines $\textsc{PnP}_{M,F,S}$. Although we do not discuss the runtime of the machines explicitly, we require the machines M, F and S to be efficient and to not blow up the runtime of $\textsc{PnP}_{M,F,S}$. The runtime is measured (and has to be polynomial) in terms of the first (omitted) runtime parameter, not in terms of other inputs they may receive from the protocol.

Theorem 2.4.2. *Every Plug'n'Play adversary class is single-challenge reducible as in Definition 2.4.3.*

Proof. Let M, F and S be three arbitrary $\mathtt{PPT}$ machines for a PnP adversary class $\textsc{PnP}_{M,F,S}$. We show that that $\textsc{PnP}_{M,F,S}$ has the properties from Definition 2.4.3.

- **Alpha-Renaming:** The Plug'n'Play adversary class $\textsc{PnP}_{M,F,S}$ uses challenge tags Ψ exactly for one purpose: for handling one instance of a machine M per challenge tag, obliviously to the tag itself. The behavior of the machines F and M and S is not affected by Ψ (simply because they never see them). Thus, for any permutation on the challenge tags, the adversary class will have exactly the same behavior.
- **Reliability:** Note that the challenger never initializes challenges, i.e., will never send any message containing $(\textsc{Ch}, \Psi)$ before receiving a message containing $(\textsc{Ch}, \Psi)$. By construction, $\textsc{PnP}_{M,F,S}$ only sends messages with $\mathsf{ID} = (\textsc{Ch}, \Psi)$ when M_Ψ outputs them (F cannot communicate with the challenger and S cannot send a message $\mathtt{Challenge}$). However, $\textsc{PnP}_{M,F,S}$ invokes M_Ψ only if it (before) receives a message with $\mathsf{ID} = (\textsc{Ch}, \Psi)$. Since the challenger cannot send a message with $\mathsf{ID} = (\textsc{Ch}, \Psi)$ before receiving a message with $\mathsf{ID} = (\textsc{Ch}, \Psi)$, $\textsc{PnP}_{M,F,S}$ only initializes and invokes M_Ψ if the adversary has send a message with $ID = (\textsc{Ch}, \Psi)$ before. Thus, $\textsc{PnP}_{M,F,S}$ is reliable.
- **Simulatability:** We construct the following *generic simulator* $\textsc{Sim}$ on input z and the anonymity notion α:
  - $\textsc{Sim}$ initialized a set of known IDs $\mathcal{I} := \emptyset$ and a set of challenge states $\mathsf{States} = [\mathtt{FRESH}, \ldots, \mathtt{FRESH}]$ for all γ possible challenges, where γ is directly infered from z.
  - Whenever $\textsc{Sim}$ receives any message $m = (\ldots, \mathsf{ID})$, where $\mathsf{ID} = (\textsc{Ch}, \Psi)$ with $z_\Psi = \mathtt{don't\ simulate}$, $\textsc{Sim}$ forwards the message.
  - Whenever $\textsc{Sim}$ receives any message $m = (\ldots, \mathsf{ID})$, where $\mathsf{ID} = (\textsc{Ch}, \Psi)$ with $z_\Psi = \mathtt{simulate}$ (for the bit b_Ψ), behave as follows:
    - If M_Ψ was not initialized so far (i.e., if a message with $\mathsf{ID} = (\textsc{Ch}, \Psi)$ is received for the first time), initialize a new instance of the machine M as M_Ψ. Moreover, draw a new nonce z and add (Ψ, z) to $\mathcal{I}$.
    - Then / otherwise simulate M_Ψ on m (replace ID with $\perp$), until it outputs a message m' and relay m', depending on its structure:
      - If $m' = \mathtt{Observations} = (t, \mathsf{ID}')$, relay m' to A, but replace the ID' with $(\textsc{Ch}, \Psi)$.
      - If $m' = \mathtt{Input} = (\mathsf{S}, \mathsf{R}, m'', \mathsf{ID})$, relay m' to $\textsc{Ch}$, but replace ID' with (A, z).

· If $m' = \texttt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, m'', \mathsf{ID}')$, then compute the challenge as $(\mathsf{state}', \mathsf{S}^*, \mathsf{R}^*) := \alpha(\mathsf{States}[\Psi], \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b_\Psi)$, set the corresponding state as $\mathsf{States}[\Psi] := \mathsf{state}'$ and send $\texttt{Input} = (\mathsf{S}^*, \mathsf{R}^*, m'', (\mathrm{A}, \mathsf{z}))$ to CH.

We consequently show that the general simulator SIM satisfies all conditions from Definition 2.4.3, Item 3.

– **Item 3a:** Let $z \in \{0,1\}^{2\gamma}$ be any simulator index and $i \in \{1, \ldots, \gamma\}$, s.t., $z_i = \texttt{simulate}$. Then by construction SIM_z intercepts all messages with $\mathsf{ID} = (\mathrm{CH}, i)$ and feeds them into a simulated machine M_i. Whenever this machine outputs a message $\texttt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, m, \mathsf{ID}')$, SIM replaces it by $\texttt{Input} = (\mathsf{S}^*, \mathsf{R}^*, m, (\mathrm{A}, \mathsf{z}))$ for some nonce z, where $(\_, \mathsf{S}^*, \mathsf{R}^*) := \alpha(\mathsf{States}[\Psi], \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b_\Psi)$. Furthermore, SIM never introduces messages of the type $\texttt{Challenge} = (\ldots, (\mathrm{CH}, j))$ for any j: it only forwards messages of this form if they are sent by A and if $z_j = \texttt{don't simulate}$. Thus, SIM never sends a message $\texttt{Challenge} = (\ldots, (\mathrm{CH}, i))$.

– **Item 3b:** The simulator $\mathrm{SIM}(z_{\texttt{real}}, \alpha)$ forwards all messages it receives. It never intercepts messages with $\mathsf{ID} = (\mathrm{CH}, i)$ for any i and never intercepts any messages with $\mathsf{ID} = (\mathrm{A}, \mathsf{z})$ for any z. Thus, $\mathrm{SIM}(z_{\texttt{real}})$ does not modify the communication and consequently, the games are information theoretically indistinguishable.

– **Item 3c:** We now show that our simulator correctly simulates challenges (if the bit within its index is correct). Let γ be the number of allowed challenges, let α be the anonymity notion and, let b be the bit of the challenger and let $z, z' \in \{0,1\}^{2\gamma}$ such that

$$\forall i \in \{1, \ldots, n\} \text{ s.t. } z_i \neq z'_i.\ (z_i = \texttt{simulate} \Rightarrow b_i = b) \\ \wedge (z'_i = \texttt{simulate} \Rightarrow b'_i = b).$$

We now show that the gamed $\texttt{AC-Sim}_{\mathrm{SIM}(z,\alpha)}(\Pi, \alpha, \gamma, b)$ and $\texttt{AC-Sim}_{\mathrm{SIM}(z',\alpha)}(\Pi, \alpha, \gamma, b)$ (as in Definition 2.4.3) are indistinguishable.

We begin with noticing that both our generic simulator and the adversary class do not share state between different ID's: Their behavior on messages $(\_, \mathsf{ID})$ does not in any way depend on any communication or computation on messages $(\_, \mathsf{ID}')$ for any other ID', except for F possibly keeping state. Consequently, we can analyze their behavior for each ID separately for a state state_F of F.

We furthermore assume that there are indexes $i \in \{1, \ldots, n\}$ s.t. $z_i \neq z'_i$. Let i be such an index and let us assume w.l.o.g., that $z_i = \texttt{simulate}$. The difference in behavior covers exactly the messages with $\mathsf{ID} = (\textsc{Ch}, i)$ by A.[2] To simplify the proof we can thus assume that there is exactly one difference (for index i).

Whenever a message with $\mathsf{ID} = (\textsc{Ch}, i)$ is sent by A to $\textsc{Sim}_{z'}$, the simulator simply forwards the message to $\textsc{PnP}_{M,F,S}$, which, in turn, feeds it to the machine M_i. Analogously, $\textsc{Sim}_z$ initializes a machine M_i and feeds it all such messages. Whenever in the game with $\textsc{Sim}_{z'}$ the challenger replies to the adversary with a message $\texttt{Observations} = (x, (\textsc{Ch}, i))$, $\textsc{PnP}_{M,F,S}$ also feeds it to M_i. If these messages instead (as in the game with $\textsc{Sim}_z$) have the ID (A, z) for any nonce z, then $\textsc{PnP}_{M,F,S}$ simply forwards them and $\textsc{Sim}_z$ feeds them into its machine M_i.

We proceed by an inductive proof over all messages sent to and received by M_i.

The first message must originate from A, as the challenger never initializes challenges (i.e., it never sends messages $(\textsc{Ch}, i)$ without receiving such messages before). Before A sends such a message, the two considered games $\texttt{AC-Sim}_{\textsc{Sim}(z,\alpha)}(\Pi, \alpha, \gamma, b)$ and $\texttt{AC-Sim}_{\textsc{Sim}(z',\alpha)}(\Pi, \alpha, \gamma, b)$ are information theoretically indistinguishable (the simulators do not behave differently). For this first message m, $\textsc{Sim}_z$ now creates a fresh nonce z for translating the challenge ID and then initializes the machine M_i with m. Analogously, $\textsc{Sim}_{z'}$ simply forwards the message to $\textsc{PnP}_{M,F,S}$ that initializes M_i with m. Both machines will behave indistinguishably, as they are initialized with the same message. The games only differ if M_i outputs a message $\texttt{Challenge} = (\ldots, \mathsf{ID}')$. In $\texttt{AC-Sim}_{\textsc{Sim}(z',\alpha)}(\Pi, \alpha, \gamma, b)$ this message is sent to $\textsc{Ch}$ which, since this is a challenge message, applies the anonymity function α to it (using challenge bit b) and forwards the resulting message to the protocol. In $\texttt{AC-Sim}_{\textsc{Sim}(z,\alpha)}(\Pi, \alpha, \gamma, b)$, the simulator applies α to it (using the same challenge bit b) and forwards the resulting message as $\texttt{Input} = (\mathsf{S}^*, \mathsf{R}^*, m, (\mathrm{A}, \mathsf{z})$ to $\textsc{PnP}_{M,F,S}$, which forwards the message to $\textsc{Ch}$ ($\textsc{PnP}_{M,F,S}$ does not modify input messages), which, again (since this is an input message) directly forwards it to the protocol.[3]

Consider any later point in the games, where so far the machines M_i behaved indistinguishably and the games were also indistinguishable. If suddenly the machines M_i behaved in a distinguishable manner, they would entail a distinguisher on the aforementioned indistinguishability. Again, the games

[2]Technically, $\textsc{Sim}_z$ translates the communication regarding this challenge ID to (A, z) when communicating to the challenger, for a random nonce z. The messages with $\mathsf{ID} = (\mathrm{A}, \mathsf{z})$ could thus also be affected.

[3]Technically, the challenger handles sessions in both cases, once for ID $(\textsc{Ch}, i)$, once for ID (A, z). Its behavior may only differ if the real adversary sends a message with $\mathsf{ID} = (\mathrm{A}, \mathsf{z})$ before, or after the challenge, but since z is a random nonce, this occurs with negligible probability only.

only are structurally different if M_i outputs a message $\texttt{Challenge} = (\ldots)$, but this message will be transformed to a sender S^*, a recipient R^* and a message m^* for the protocol Π that is the same for both games and for this tuple $(\mathsf{S}^*, \mathsf{R}^*, m^*)$ the challenger will handle protocol sessions in exactly the same way as long as there is no collision of the ID (A, z) with IDs generated by A, which, however, occurs with negligible probability only.

This concludes the proof. □

Remark 2.4.1. *It is, theoretically, possible to compose adversary classes by nesting them into each other or by sequentially applying them. However, such a nested composition of adversary classes is then not necessarily single-challenge reducible in terms of Definition 2.4.3: The fact that answers from the challenger either are filtered twice (for adversary IDs and consequently for answers to the simulator) or input to a challenge-machine M in between the two filters can cause the simulation to fail. We could restrict the filter machine F to only apply to challenge messages, which would make such a composition possible, but this would also restrict the versatility of our* PnP *adversary class.*

We rather focus on an easy-to-use PnP *adversary class in which the machines themselves can enforce combinations of different restrictions for the adversary.*

Example 2.4.3. *The following simple examples for adversary classes show the expressiveness of our* PnP *adversary class.*

- *Fixed-Sender-Recipient adversaries: We want the adversary to only select two predefined senders S_0, S_1 and/or two predefined recipients R_0, R_1 for its challenges. Consequently, we set F to the identity machine (simply forwarding all inputs) and define M as follows: M forwards all messages of the form $\texttt{Input} = (\mathsf{S}, \mathsf{R}, m, \mathsf{ID})$ and replaces all messages $\texttt{Challenge} = (\mathsf{S}_x, \mathsf{S}_y, \mathsf{R}_x, \mathsf{R}_y, m_0, m_1, \mathsf{ID})$ by $\texttt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, m_0, m_1, \mathsf{ID})$.*

- *Static compromisation adversaries: We set M to the identity machine (simply forwarding all inputs), F to a compromisation blocking machine, blocking all messages $(\texttt{compromise}, \ldots)$ from* A *to Π and S to the following machine: S communicates with the adversary about its compromisation strategy and then sends messages $(\texttt{compromise}, P)$ for all respective parties P that* A *wants and is allowed to compromise.*

- *More elaborate compromisation: Whenever F is initialized with the adversary's compromisation strategy (N, L), F checks and/or directly modifies the sets and outputs (N', L'), thereby enforcing arbitrary compromisation restraints. One versatile and useful example (that we also discuss in detail later) is a* budget adversary*, where F is parametric in a budget B and a cost function f that checks* *whether* $\sum_{\mathsf{n} \in N \cap \mathcal{N}} f(\mathsf{n}) < B$ *and if so forwards* $(N, \emptyset)$*, i.e., F allows the adversary to compromise Tor nodes up to the budget B and does not allow the adversary to compromise connections between Tor entities.*

- *Combined adversary class: We later combine all three mentioned adversary classes to calculate anonymity guarantees for eavesdroppers that statically compromise Tor nodes and wiretap communications between them. Consequently, we put the restrictions mentioned for the "more elaborate compromisation" adversary class on S instead of on F and block all compromisation messages from F. Moreover we restrict the challenges by using the machine M from the "Fixed-Sender-Recipient" adversary to calculate guarantees for specific anonymity scenarios.*

Chapter 3

Impact of Passive Adversaries on Tor

In this chapter we discuss the possible observations that any passive adversary against Tor can make for any given communication. To this end, we first formalize the intuitive description of observation points we introduced in Section 1.2.4. Subsequently we present a formalization of Tor in the universal composability framework and show that anonymity guarantees shown for the idealization carry over to the cryptographic instantiation and, under reasonable assumptions, for Tor's implementation. Based on the concept of observations and the ideal functionality of the UC formalization, we present a simplified variant of the AnoA challenge-response game. We show that it suffices to analyze the advantage of eavesdroppers in this simplified game to derive anonymity guarantees for the ideal functionality and, thus, for Tor's actual implementation.

3.1 Observation Points of Tor Circuits

Observing any Tor entity or connection between Tor entities involved in a Tor circuit enables the adversary to draw certain conclusions about the sender and/or the recipient of the circuit, thereby reducing their degree of anonymity.

Some of these conclusions are straightforward and result in immediate deanonymization: if the guard node can be observed, the sender is trivially deanonymized as the origin of the communication; similarly, observing the exit node unveils the recipient as the destination of the communication. Existing papers are typically limited to such all-or-nothing observations. However, additional conclusions can be drawn when corrupting *any* Tor entity or connection between them, and these conclusions are no less influential. For instance, if the adversary observes or knows that a recipient requires a specific TCP port, all exit nodes can be excluded that do not support this port, and hence, any communication that involves excluded exit nodes cannot involve this recipi-

Guard Node

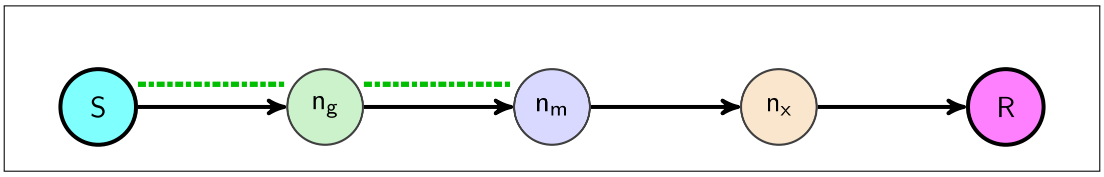


Figure 3.1: The guard node $\mathsf{n_g}$ of a circuit $\mathsf{C} = (\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}) \in \mathsf{Circuits}$ can observe the sender S and the middle node $\mathsf{n_m}$.

ent. Moreover, excluding exit nodes influences the probability of which nodes are being selected as guard or middle nodes in this circuit by Tor's path selection algorithm: The selection takes so-called family relationships and further constraints into account. Technically, this means that the a-priori probability distribution over circuits induced by Tor's path selection algorithm is now replaced by an a-posteriori distribution that is conditioned on the observations of the adversary. This enables the adversary to draw further conclusions and to thereby reduce anonymity.

3.1.1 Observations

Now we define which observations an adversary is able to make if certain nodes are considered corrupted. We consider a distinguished symbol, denoted $\perp$, that reflects that an observation at a certain position in the Tor circuit cannot be made. Further, we define the overall impact on anonymity if a given set of nodes is considered under adversarial control.

Definition 3.1.1 (Observations and Circuits)**.** *For a set of senders $\mathcal{S}$, a set of recipients $\mathcal{R}$, and the set of all Tor nodes $\mathcal{N}$, we define the set of* Tor circuits $\mathsf{Circuits}$ *between these senders and recipients as* $\mathsf{Circuits}_{\mathcal{S},\mathcal{R},\mathcal{N}} := \mathcal{S} \times \mathcal{N}^3 \times \mathcal{R}$ *and the set of* observations *as* $\mathcal{O}_{\mathcal{S},\mathcal{R},\mathcal{N}} := (\mathcal{S} \cup \{\perp\}) \times (\mathcal{N} \cup \{\perp\})^3 \times (\mathcal{R} \cup \{\perp\})$ *for the distinguished symbol $\perp$. We omit the subscripts if they are clear from the context and, hence, write* $\mathsf{Circuits}$ *and $\mathcal{O}$.*

For a set $N \subseteq \mathcal{E}$ of Tor entities, and a set $L \subseteq \mathcal{E}^2$ of connections between Tor entities, we now define the observations $\textsc{obs}[N, L](\mathsf{C}) \in \mathcal{O}$ made by N and L within a considered Tor circuit C. Intuitively, whenever a node $\mathsf{n} \in N$ is part of the circuit C, then this node as well as its successor and predecessor can be identified. If $\mathsf{n} \in \mathcal{S}$, then the sender and the guard node can be identified; if $\mathsf{n} \in \mathcal{R}$, the exit node and the recipient can be identified.

Formal Definition of Observations per Tor Circuit Based on the possible observation points (c.f. Figures 3.1 to 3.5), we define the observations that any combination

Middle Node

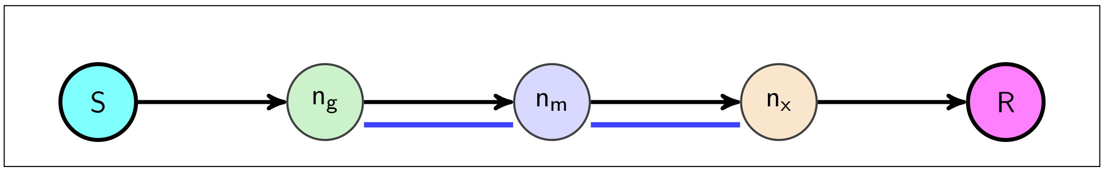


Figure 3.2: The middle node $\mathsf{n_m}$ of a circuit $\mathsf{C} = (\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}) \in \mathsf{Circuits}$ can observe the guard node $\mathsf{n_g}$ and the exit node $\mathsf{n_x}$.

Exit Node

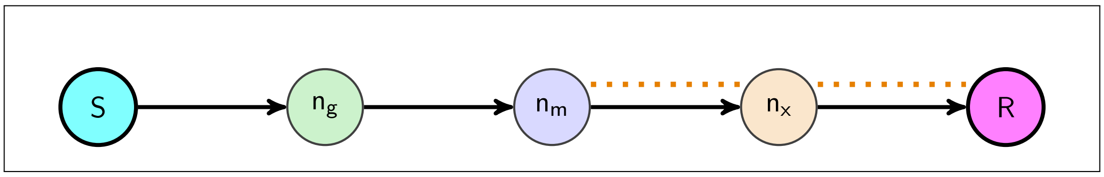


Figure 3.3: The exit node $\mathsf{n_x}$ of a circuit $\mathsf{C} = (\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}) \in \mathsf{Circuits}$ can observe the middle node $\mathsf{n_m}$ and the exit node R.

Sender and Recipient observations

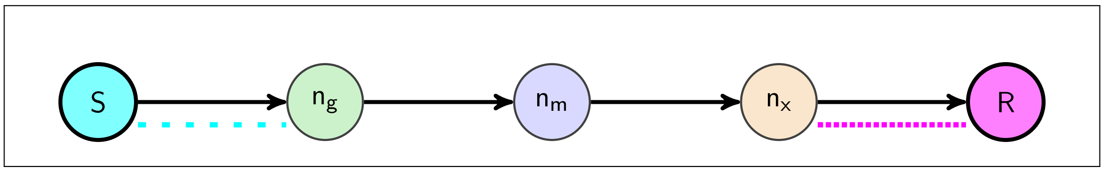


Figure 3.4: The sender S of a circuit $\mathsf{C} = (\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}) \in \mathsf{Circuits}$ can observe the guard node $\mathsf{n_g}$; the recipient R of C can observe the exit node $\mathsf{n_x}$.

Compromised Connections between Tor Entities

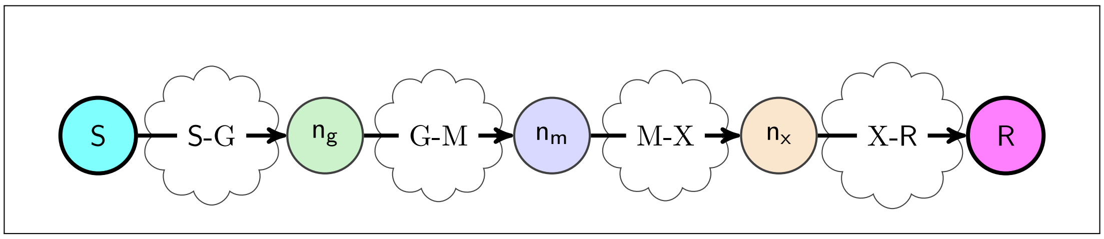


Figure 3.5: An observer on the connection between two Tor entities e_1 and e_2 can, naturally, observe both e_1 and e_2. In this graph we show all such observation points between Tor entities.

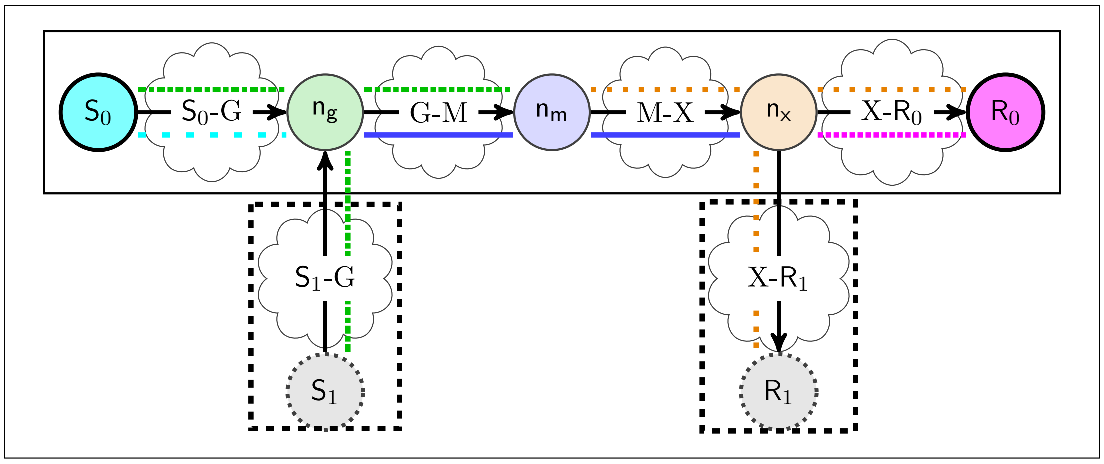


Figure 3.6: Comprehensive overview over the possible observations for a Tor circuit in a challenge, where one of two senders $\mathsf{S}_0, \mathsf{S}_1$ sends a message to one of two recipients $\mathsf{R}_0, \mathsf{R}_1$. We portray all observation points, i.e., all Tor entities of the circuit and all connections between them.

of Tor entities N and connections between Tor entities L can make for any given circuit.

Definition 3.1.2 (Circuit Observations)**.** *For a set of senders $\mathcal{S}$, a set of recipients $\mathcal{R}$, and for $N \subseteq \mathcal{E}$, $L \subseteq \mathcal{E}^2$, the* circuit observation *of N and L is a function* $\textsc{obs}[N, L] : \mathsf{Circuits} \to \mathcal{O}$ *and is defined as follows. For* $\mathsf{C} = (\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}) \in \mathsf{Circuits}$*, we have* $\textsc{obs}[N, L](\mathsf{C}) := (n_1, \dots n_5)$ *with*

- $n_1 := \mathsf{S}$ *if* $\{\mathsf{S}, \mathsf{n_g}\} \cap N \neq \emptyset$ *or* $(\mathsf{S}, \mathsf{n_g}) \in L$*; otherwise* $n_1 := \perp$.
- $n_2 := \mathsf{n_g}$ *if* $\{\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}\} \cap N \neq \emptyset$ *or* $\{(\mathsf{S}, \mathsf{n_g}), (\mathsf{n_g}, \mathsf{n_m})\} \cap L \neq \emptyset$*; otherwise* $n_2 := \perp$.
- $n_3 := \mathsf{n_m}$ *if* $\{\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}\} \cap N \neq \emptyset$ *or* $\{(\mathsf{n_g}, \mathsf{n_m}), (\mathsf{n_m}, \mathsf{n_x})\} \cap L \neq \emptyset$*; otherwise* $n_3 := \perp$.
- $n_4 := \mathsf{n_x}$ *if* $\{\mathsf{n_m}, \mathsf{n_x}, \mathsf{R}\} \cap N \neq \emptyset$ *or* $\{(\mathsf{n_m}, \mathsf{n_x}), (\mathsf{n_x}, \mathsf{R})\} \cap L \neq \emptyset$*; otherwise* $n_4 := \perp$.
- $n_5 := \mathsf{R}$ *if* $\{\mathsf{n_x}, \mathsf{R}\} \cap N \neq \emptyset$ *or* $(\mathsf{n_x}, \mathsf{R}) \in L$*; otherwise* $n_5 := \perp$.

We call (N, L) the observation points *of* $\textsc{obs}$.

3.2 Tor's Universal Composability Protocol

In this section we show that it suffices to analyze Tor in an abstract fashion modeled by an ideal functionality in the universal composability (UC) framework. Under reasonable assumptions, these analyses are equivalent to guarantees for Tor's implementation against statically corrupting, passive adversaries, up to a negligible factor. To this end, we briefly review the Tor model of Backes, Goldberg, Kate and Mohammadi, which they defined in the *universal composability* (UC) framework of Canetti [**UC**, 15]. We show that differential-privacy like guarantees (such as *IND-CDP*) shown for an ideal functionality also hold for its cryptographic instantiation, which provides a slight relaxation of classical UC proofs.

3.2.1 Overview of Tor in UC

The UC framework allows for a modular analysis of security protocols. In the UC framework, the security of a protocol is defined by comparing an execution of the protocol with a setting in which all parties have a direct and private connection to a trusted machine that provides the desired functionality. As an example consider an authenticated channel between two parties Alice and Bob. In the real world Alice calls a protocol that signs the message m to be communicated. She then sends the signed message over the network and Bob verifies the signature. In the setting with a trusted machine T, however, we do not need any cryptographic primitives: Alice sends the message m directly to T. T in turn sends m to Bob, who trusts T and can be sure that the message is authentic. The trusted machine T is called the *ideal functionality.*

We briefly review the formal UC definition (an ideal functionality $\mathcal{F}_{\mathrm{OR}}$) for the onion routing (OR) network and its cryptographic instantiation [15]. $\mathcal{F}_{\mathrm{OR}}$ presents the OR definition in the message-based state transitions form, and defines sub-machines for all OR nodes in the ideal functionality. These sub-machines share a memory space in the functionality for communicating with each other. $\mathcal{F}_{\mathrm{OR}}$ assumes an adversary who might possibly control all communication links and destination servers, but cannot view or modify messages between uncompromised parties due to the presence of secure and authenticated channels between the parties. In $\mathcal{F}_{\mathrm{OR}}$ these secure channels are realized by having each party store their messages in the shared memory, and create and send corresponding handles $\langle P, P_{\mathtt{next}}, h\rangle$ through the network. Here, P and $P_{\mathtt{next}}$ are the sender and the recipient of a message respectively and h is a handle, or pointer, for the message in the shared memory. Only messages that are visible to compromised parties are forwarded to A.

Tor sets a time limit (of ten minutes) for each established circuit. However, the UC framework does not provide a notion of time. $\mathcal{F}_{\mathrm{OR}}$ models such a time limit by only allowing a circuit C to transport at most a constant number (say `ttl-count`) of messages. We could implement this restriction via the adversary class. However, for ease of notation and since the number of messages sent over a circuit does not impact anonymity, we do not consider this technical detail in our analysis. Instead, we assume that there is a bound on the number of messages the adversary sends per circuit and we set `ttl-count` to this bound.

3.2.2 Modifications

We base our model of Tor on our previous work that models Tor as a UC protocol Π_{OR} [15]. Π_{OR} is based on Tor's specification and accurately models the key exchange, the circuit establishment, and the process of relaying message cells over Tor.

The cryptographic protocol Π_{OR} [15] abstracts from Tor's path selection by considering a uniform path selection. In our work, we use an extension of Π_{OR}, where instead of the uniform path selection, we treat Tor's path selection as a distribution depending on the sender and recipient of the circuit. This extension of Π_{OR}, which we call Π'_{OR}, solely extends the path selection algorithm in Π_{OR} and leaves everything else untouched. We accordingly extend the ideal functionality $\mathcal{F}_{\text{OR}}$ from [15], which abstracts from all cryptographic operations in Π_{OR}, with Tor's actual path selection. We call the extension of the ideal functionality $\mathcal{F}_{\text{OR}}$ with Tor's actual path selection $\mathcal{F}'_{\text{OR}}$. Since Π_{OR} and $\mathcal{F}_{\text{OR}}$ both execute the same path selection algorithm, the UC realization proof for Π_{OR} and $\mathcal{F}_{\text{OR}}$ applies verbatim to Π'_{OR} and $\mathcal{F}'_{\text{OR}}$ (Proposition 3.2.1).

In [15] *secure OR modules* are defined, which comprise a one-way authenticated key exchange protocol, and secure encryption and decryption operations. Moreover, that work uses a network functionality $\mathcal{F}_{\text{NET}_q}$ for modeling partially global network adversaries that compromise at most q links, a key registration functionality $\mathcal{F}_{\text{REG}}$ and a functionality for a confidential and mutually authenticated channel (for modeling HTTPS connections) $\mathcal{F}_{\text{SCS}}$.

We rewrite the environment s.t. the adversary does not per se control all recipients. Only if the recipient has received a message `compromise`, the adversary can observe all incoming messages. If a message is sent to a recipient R from a Tor node n and the adversary has sent $\mathtt{observe}(\mathsf{n}, \mathsf{R})$, then the adversary also learns the message.

Proposition 3.2.1. *[UC Realization of Tor c.f.[15]] If Π'_{OR} uses secure OR modules M, then with the global functionality $\mathcal{F}_{\text{NET}_q}$ the resulting protocol Π'_{OR} in the $\mathcal{F}_{\text{REG},\text{SCS}}$-hybrid model securely realizes the ideal functionality $\mathcal{F}'_{\text{OR}}$ for any q.*

3.2.3 Differential Privacy Style Guarantees in UC

In this section, we prove that differential privacy (and therefore also *IND-CDP*) are preserved under realization. This preservation allows for an elegant crypto-free anonymity proof for cryptographic AC protocols with AnoA.

Security in the UC framework is defined as follows: A real protocol is secure if an execution of this real protocol is indistinguishable from an execution of the corresponding ideal functionality. Here, indistinguishability is defined in terms of binary random variables which represent the output of the probabilistic real protocol and ideal functionality.

Definition 3.2.1 (Indistinguishability (Canetti))**.** *Two binary distribution ensembles X and Y are indistinguishable, denoted $X \approx Y$ if for every $c \in \mathbf{N}$ there is a $\eta_0 \in \mathbf{N}$*

such that for all $\eta > \eta_0$ and all x we have that

$$|Pr[X(\eta, x)] = 1 - Pr[Y(\eta, x)] = 1| < \delta' = \eta^{-c}$$

The Real World For the process in the real world we introduce the random variable $\texttt{Real}_{\Pi,A,D}(\eta, x)$ which captures the interaction of a protocol Π with an adversary A, observed by a distinguisher D. $\texttt{Real}_{\Pi,A,D}$ will denote the ensemble of all those distributions. Note that as we try to argue about *IND-CDP*, our input x will be a tuple of inputs (x_0, x_1).

The Ideal World Similarly, we introduce the random variable $\texttt{Ideal}_{\mathcal{F},S,D}(\eta, x)$ which captures the interaction of an ideal functionality $\mathcal{F}$, a simulator S and the distinguisher. $\texttt{Ideal}_{\mathcal{F},S,D}$ will again denote the ensemble of such random variables.

If the execution of a real protocol is indistinguishable from the execution of the corresponding ideal functionality, we say that the protocol UC-realizes the ideal functionality.

Definition 3.2.2 (Realization in UC)**.** *A protocol* Π *UC-realizes an ideal functionality F if for every* PPT *adversary* A *of* Π *there exists a* PPT *simulator S sucht that for every* PPT *distinguisher D it holds that*

$$\texttt{Real}_{\Pi,A,D} \approx \texttt{Ideal}_{\mathcal{F},S,D}$$

Preservation of Differential Privacy The UC-realization does not only allow us to prove the security of the real protocol given a trivially secure ideal functionality, but also allows to lift security guarantees proven for the realized protocol to the realizing protocol: given the realization of a (ϵ, δ)-differentially-private ideal functionality by a protocol Π, we get differential privacy for Π as well. This result is motivated by the ideas presented in the result of integrity property conservation by simulation-based indistinguishability shown by Backes and Jacobi [7, Thm. 1].

Theorem 3.2.1. *If* Π *UC-realizes an* (ϵ, δ)*-dp functionality* $\mathcal{F}$ *then* Π *is* (ϵ, Δ)*-dp with* $\Delta = \delta + \delta'$ *for some negligible value* δ'*.*

Proof. Given an (ϵ, δ)-dp functionality $\mathcal{F}$, assume Π UC-realizes $\mathcal{F}$, but Π is not (ϵ, Δ)-dp, i.e. there exist an adversary A and two inputs x_0 and x_1 s.t.

$$Pr[A(\Pi(x_1)) = 1] > e^{\epsilon} Pr[A(\Pi(x_0)) = 1] + \Delta$$

such that $\Delta \geq \delta + \delta'$ for a non negligible value δ'. We construct the following PPT distinguisher D that uses A in order to separate Π from $\mathcal{F}$:

1. choose $b \xleftarrow{R} \{0, 1\}$ uniformly at random
2. send x_b through the network

3. depending on the output b^* of the adversary:
   (a) if the adversary returns $b^* = b$ output 1 (most likely (Π, A) was observed).
   (b) otherwise output 0 (most likely $(\mathcal{F}, S)$ was observed).

We now bound the probabilities $\Pr[\texttt{Real}_{\Pi,A,D}(\eta, (x_0, x_1)) = 1]$ and $\Pr[\texttt{Ideal}_{\mathcal{F},S,D}(\eta, (x_0, x_1)) = 1]$ as required for the lemma. We will use the expressions $A^b_{\texttt{Real}_{\Pi,D}}$ and $S^b_{\texttt{Ideal}_{\mathcal{F},D}}$ to denote the output of the adversary and simulator respectively during the execution after the distinguisher decided on a specific $b \in \{0, 1\}$. Using the assumption that Π is not (ϵ, Δ)-differentially private, the first expression computes to

$$\begin{aligned}
& \Pr[\texttt{Real}_{\Pi,A,D}(\eta, (x_0, x_1)) = 1] \\
= \ & \Pr[A^b_{\texttt{Real}_{\Pi,D}} = b] \\
= \ & \Pr[A^1_{\texttt{Real}_{\Pi,D}} = 1] \cdot \Pr[\text{D chooses } x_1] + \Pr[A^0_{\texttt{Real}_{\Pi,D}} = 0] \cdot \Pr[\text{D chooses } x_0] \\
= \ & \Pr[A(\Pi(x_1)) = 1] \cdot \Pr[b = 1 | b \overset{R}{\leftarrow} \{0, 1\}] + \Pr[A(\Pi(x_0)) = 0] \cdot \Pr[b = 0 | b \overset{R}{\leftarrow} \{0, 1\}] \\
= \ & \frac{1}{2} \left(\Pr[A(\Pi(x_1)) = 1] + \Pr[A(\Pi(x_0)) = 0]\right) \\
> \ & \frac{1}{2} \left(\Pr[A(\Pi(x_0)) = 1] e^\epsilon + \Delta + \Pr[A(\Pi(x_0)) = 0]\right) \\
= \ & \frac{1}{2} \left(\Pr[A(\Pi(x_0)) = 1] e^\epsilon + \Delta + 1 - \Pr[A(\Pi(x_0)) = 1]\right) \\
= \ & \frac{1}{2} \left((e^\epsilon - 1) \Pr[A(\Pi(x_0)) = 1] + \Delta + 1\right) \\
\geq \ & \frac{1}{2} (1 + \Delta)
\end{aligned} \tag{3.1}$$

Using the (ϵ, δ)-differential privacy of $\mathcal{F}$, the second expression can be bound as follows

$$\begin{aligned}
& \Pr[\texttt{Ideal}_{\mathcal{F},S,D}(\eta, (x_0, x_1)) = 1] \\
= \ & \Pr[S^b_{\texttt{Ideal}_{\mathcal{F},D}} = b] \\
= \ & \Pr[S^1_{\texttt{Ideal}_{\mathcal{F},D}} = 1] \cdot \Pr[\text{D chooses } x_1] + \Pr[S^0_{\texttt{Ideal}_{\mathcal{F},D}} = 0] \cdot \Pr[\text{D chooses } x_0] \\
= \ & \Pr[S(\mathcal{F}(x_1)) = 1] \cdot \Pr[b = 1 | b \overset{R}{\leftarrow} \{0, 1\}] + \Pr[S(\mathcal{F}(x_0))) = 0] \cdot \Pr[b = 0 | b \overset{R}{\leftarrow} \{0, 1\}] \\
= \ & \frac{1}{2} \left(\Pr[S(\mathcal{F}(x_1)) = 1] + \Pr[S(\mathcal{F}(x_0)) = 0]\right) \\
\leq \ & \frac{1}{2} \left(\Pr[S(\mathcal{F}(x_0)) = 1] e^\epsilon + \delta + \Pr[S(\mathcal{F}(x_0)) = 0]\right) \\
= \ & \frac{1}{2} \left((1 - \Pr[S(\mathcal{F}(x_0)) = 0]) e^\epsilon + \delta + \Pr[S(\mathcal{F}(x_0)) = 0]\right) \\
= \ & \frac{1}{2} \left(e^\epsilon + (1 - e^\epsilon) \Pr[S(\mathcal{F}(x_0)) = 0] + \delta\right) \\
\leq \ & \frac{1}{2} \left(e^\epsilon + 1 - e^\epsilon + \delta\right) \\
= \ & \frac{1}{2} (1 + \delta)
\end{aligned} \tag{3.2}$$

(3.3)

Putting Equations 3.1 and 3.2 together, we then get

$$\Pr[\mathtt{Real}_{\Pi,A,D}(\eta,(x_0,x_1))=1]-\Pr[\mathtt{Ideal}_{\mathcal{F},S,D}(\eta,(x_0,x_1))=1]>\frac{1}{2}(\Delta-\delta)=\frac{1}{2}\delta'$$

for a non negligible value $\frac{\delta'}{2}$. Hence D is a PPT machine distinguishing (A,Π) from $(S,\mathcal{F})$ with more than negligible probability, contradicting the UC-realization of $\mathcal{F}$ by Π (cf Definition 3.2.2). Therefore our initial assumption is wrong and Π is (ϵ,Δ)-differentially private. □

In the same vein as above, we can also show that *IND-CDP* is preserved by UC realization. As a consequence of this result, it suffices to apply AnoA to ideal functionalities: transferring the results to the real protocol weakens the anonymity guarantees only by a negligible amount.

Corollary 3.2.1. *Let* Π *be* (ϵ,δ)*-IND-CDP and* Π *be a protocol. If* Π *UC-realizes* Π *then* Π *is* (ϵ,Δ)*-IND-CDP with* $\Delta=\delta+\delta'$ *for some negligible value* δ'*.*

The above result, in combination with an ideal functionality for an AC protocol, is useful for analyzing the AC protocol with respect to our strong anonymity definitions.

Formalization of Tor Entities Senders can create Tor circuits by sending a message $(\mathtt{create\ circuit},\mathsf{C})$ to Tor nodes, where $\mathsf{C}\in\mathsf{Circuits}$ describes the Tor nodes that should be used for the circuit creation. Subsequently, Tor nodes receive messages `extend circuit` and reply with `created` if they successfully extended the circuit. After a circuit was created, it can be used to send messages that are subsequently propagated by the Tor nodes.

Recipient Wrapper Our formalization of Tor is parametric in a stateless machine Rec that computes responses of recipients. The machine needs to be stateless to ensure that there is no indirect information flow between different recipients. As our analysis targets anonymity of IP addresses, we require that Rec is independent of its first input, i.e., the response computed by Rec does not depend on the recipient to which the message was sent. We refer to Figure 3.7 for a description of the recipient wrapper machine.

Adversary Model Their Tor model considers partially global attackers. Thus, they model the network explicitly by an explicitly defined ideal functionality $\mathcal{F}_{\text{NET}}$ that allows the adversary to compromise connections between Tor nodes up to a specific number q. The adversary can send messages $(\mathtt{compromise},\mathsf{n})$ to compromise a Tor node and messages $(\mathtt{observe},\mathsf{n},\mathsf{n}')$ to compromise connections between them. We naturally modify the compromisation mechanic by applying the same restrictions on compromisation as for the AnoA adversary. Instead of allowing subsequent compromisation up to a specific number of nodes or connections, we require the adversary to initially send two sets N and L and restrict them via adversary classes, as before. Moreover, the adversary from [15] can perform active attacks, by inserting, dropping, or modifying messages. We restrict the adversary to passive attacks and from performing adaptive compromisation by the use of an AnoA adversary class that we present in the following.

We refer to Appendix A.1 for a formal definition of $\mathcal{F}_{\mathrm{OR}}$ (which is not part of this thesis, repeated for completeness).

We construct the following Plug'n'Play adversary class:

Definition 3.2.3 (Passive UC Adversary with Transparent Challenges)**.** *Given any challenge Turing machine M' and a predicate on compromisation P, we construct a passive, statically corrupting Tor adversary class as a Plug'n'Play adversary class $\mathrm{PnP}_{M,F_{\mathtt{block}},S_P}$ as follows:*

- *The challenge machine M runs the input machine M' but enforces* transparent challenges*, i.e., whenever M' sends a message m to* Ch*, M additionally sends m to* A*.*[1]

- *The filter machine $F_{\mathtt{block}}$ blocks all messages from* A *to* Π*, thus rendering* A *passive, as it cannot influence Tor messages and reflects all messages from* Π *to* A*, thus disallowing* A *from intercepting and blocking messages. As a side effect, $F_{\mathtt{block}}$ disallows adaptive compromisation, as* A *can only compromise Tor entities in the setup phase. More specifically, if F receives a message of the form (P, Q, m) from $\mathcal{F}_{\mathrm{NET}}$, then reflect (P, Q, m) back to $\mathcal{F}_{\mathrm{NET}}$. If F receives a message of the form $\langle P_1, \dots, P_n, \mathtt{cmd}, m\rangle$ from $\mathcal{F}_{\mathrm{OR}}$, then send $(P_{n-1}, P_n, \mathtt{cmd}, m)$ to $\mathcal{F}_{\mathrm{OR}}$.* [2] *In both cases, the adversary receives a copy of the message.*

- *The setup machine S_P allows* A *to compromise Tor entities, as follows:* A *can specify a set $N \subseteq \mathcal{E}$ of compromised Tor entities and a set $L \subseteq \mathcal{E}^2$ of compromised connections between Tor entities. Then, S_P checks, whether $P(N, L) = \mathtt{true}$ and only if so, allows this compromisation. However, S_P restricts the compromisation of senders and recipients depending on the anonymity notion as follows:*
  - ***if*** *$\alpha = \alpha_{\mathsf{SA}}$, let $N := N \setminus \mathcal{S}$.*
  - ***if*** *$\alpha = \alpha_{\mathsf{RA}}$, let $N := N \setminus \mathcal{R}$.*
  - ***if*** *$\alpha = \alpha_{\mathsf{REL}}$, let $N := N \setminus (\mathcal{S} \cup \mathcal{R})$.*

[1] To clearly distinguish these messages from other messages that M' sends to A, we can begin all messages from M' to A with the bit 0 and all messages that are copies of challenger messages with the bit 1.

[2] Technically, $\mathcal{F}_{\mathrm{NET}}$ (and also $\mathcal{F}_{\mathrm{OR}}$) expect a handle h, s.t., $m = \mathtt{lookup}(h)$. However, we assume that handles can be distinguished from messages and by extending the lookup algorithm $\mathtt{lookup}$ s.t. for messages m (that are not handles), $\mathtt{lookup}(m) = m$.

S_P then sends messages $(\texttt{compromise}, \mathsf{n})$ *to* $\mathcal{F}_{\text{NET}}$ *for all* $\mathsf{n} \in N$ *and* $(\texttt{observe}, \mathsf{n}, \mathsf{n}')$ *to* $\mathcal{F}_{\text{NET}}$ *for every* $(\mathsf{n}, \mathsf{n}') \in L$.

We observe that for sender anonymity, recipient anonymity and relationship anonymity the adversary cannot improve its advantage by sending several messages `Challenge` for the same challenge tag Ψ. Consequently, it suffices to focus on challenges consisting of single messages only.

Theorem 3.2.2 (Single-message challenges and inputs suffice). *Let* $\text{PnP}_{M,F_{\texttt{block}},S_P}$ *be an adversary class as defined in Definition 3.2.3, for arbitrary machines* M' *and* S_P. *Let* $\Pi^{\texttt{ONCE}}$ *be defined as* Π*, with the difference that we set* `ttl-count` *to* 1. *Under the assumption that traffic correlation attacks are perfect, it holds that whenever there is an adversary* $\text{A} \in \text{PnP}_{M,F_{\texttt{block}},S_P}$*, s.t.,* A *reduces the anonymity of* Π *in the* (α, γ)*-IND-CDP game by* (ε, δ)*, then there is an adversary* $\text{A}' \in \text{PnP}_{M',F_{\texttt{block}},S_P,\alpha}$*, s.t.,* A′ *reduces the anonymity of* $\Pi^{\texttt{once}}$ *in the* (α, γ)*-IND-CDP game by* (ε, δ).

Proof. Note that the anonymity notions α_{SA}, α_{RA} and α_{REL} ensure that for every challenge tag, exactly one fresh Tor circuit is constructed. If, as we assume, traffic correlation is perfect, then the observations that an adversary can make for any Tor circuit do not depend on the number of messages sent over this circuit. We show that for every machine A within the adversary class, we can construct a machine B that simulates all received messages that are received because of later challenge messages or later input messages. Since the challenge machine has transparent challenges, B knows the messages m that are sent and can fully simulate them, as they use the same Tor circuit that was used by the first message of this challenge. Note that for every challenge only one Tor circuit is created. As soon as the number of messages exceeds the allowed number `ttl-count` of the original protocol, the circuit is destroyed and no further circuits are created for this session ID.

Let $\text{PnP}_{M,F_{\texttt{block}},S_P}$ be an adversary class as in Definition 3.2.3, let REC be a PPT machine that computes the output of recipients and let A be an arbitrary PPT machine (run within the adversary class wrapper machine the constructive interpretation of the adversary class, c.f., Section 2.4). We construct a PPT machine B as follows:

- Internally run A and relay all messages from A with the exceptions mentioned below.
- Whenever receiving information, store it and relay it to the adversary.
- Whenever M sends a message $\texttt{Challenge}(\_, \_, \_, \_, m, \Psi)$ to the protocol, check whether this is the first message with the respective challenge tag Ψ. If so, store all data about observed communications until receiving any other message from A or from M. Whenever receiving another message $\texttt{Challenge}(\_, \_, \_, \_, m', \Psi)$ with the same tag Ψ, replay the stored observations, but replace m by m' if it occurs and replace all handles by freshly drawn handles. Moreover, compute

```
Call to Π with input $(\mathsf{S}, \mathsf{R}, m, \textsc{sid})$ from Ch:

Upon input $(\mathsf{R}, m, \textsc{sid})$
  Send message $(\mathsf{R}, m, \textsc{sid})$ to $P_\mathsf{S}$ within $\textsc{Env}_\mathsf{S}$, as specified below.

Wrapper $\textsc{Env}_u$ for onion proxy $P_u$:

Upon input $(\mathsf{R}, m, \textsc{sid})$ from Ch
  if $(\mathsf{R}, \mathsf{C}, \textsc{sid}, t) \in \mathsf{Circuits}$ for some value $\mathsf{C}$ then
    if $t <$ ttl-count then
      send message $(\texttt{send}, \mathsf{C}, (\mathsf{R}, m))$ to $P_u$
      increase $t$+=1 by storing $(\mathsf{R}, \mathsf{C}, \textsc{sid}, t+1)$ in $\mathsf{Circuits}$
    else
      Drop the message.
    end if
  else
    $\textsc{Parties} \leftarrow \textsc{ps}(P_u, \mathsf{R})$
    send message $(\texttt{create circuit}, \textsc{Parties})$ to $P_u$
    wait for response $(\texttt{created}, \mathsf{C})$
    store $(\mathsf{R}, \mathsf{C}, \textsc{sid}, 1)$ in $\mathsf{Circuits}$
    send message $(\texttt{send}, \mathsf{C}, (r, m))$ to $P_u$
  end if

Upon input $(\texttt{received}, \mathsf{C}, m)$ from $\mathcal{F}_{\mathrm{OR}}$
  if $(\mathsf{R}, \mathsf{C}, \textsc{sid}, t) \in \mathsf{Circuits}$ for some values $\mathsf{C}$ and $\mathsf{R}$ then
    send message $(m, \textsc{sid})$ to Ch.
  end if

The recipient wrapper $\textsc{Env}_r$:

Let Rec be any stateless machine that describes the behavior of recipient, i.e., that
computes the recipient's answers to messages.

Upon message $(\mathsf{R}, m, \textsc{sid})$ from machine $P$
  Run the (stateless) machine Rec on $(\mathsf{R}, P, m, \textsc{sid})$

Upon message $(P, m, \textsc{sid})$ from recipient Rec
  send $(m, \textsc{sid})$ to the protocol machine $P$
```

Figure 3.7: Wrapper machine for the environment Env .

the response of the recipient R on message m' by running the stateless recipient machine REC and sent its output and a replay of the observations made for the first response to the adversary.

For concluding the proof, it remains to be shown that the simulation of B in the game with single messages per session is correct. More precisely, we argue that the adversary A behaves the same in the simulation of B as in the original game where it can send several messages and receive observations from $\mathcal{F}_{\mathrm{OR}}$ and $\mathcal{F}_{\mathrm{NET}}$. This follows from the definition of $\mathcal{F}_{\mathrm{OR}}$ and $\mathcal{F}_{\mathrm{NET}}$, as well as our construction of B: All messages sent through the same Tor circuit will result in the same adversarial observations (with different handles), except for the message that is sent. If the adversary controls the exit node of the circuit, the recipient, or it observes the connection between the exit node and the recipient, then it can additionally observe the message. By storing the initial observations for the first message and by replaying them with replaced handles and messages, B perfectly simulates the observations. Thus, A will behave just as in the original game and by outputting the bit b^* of A, B achieves the same advantage.

Note that as long as M decides not to send challenge messages, B cannot (and does not have to) simulate messages. Since M does not receive messages from the protocol directly, it cannot tell whether or not the protocol actually sends a message through a circuit or not and thus cannot behave differently depending on whether Π or $\Pi^{\mathtt{once}}$ is used. □

Remark 3.2.1. *Note that in contrast to its name, the filter machine $F_{\mathtt{block}}$ does not filter or block any message from the protocol to the adversary. If $F_{\mathtt{block}}$ could modify messages from the protocol, then at least in general, the theorem would not hold: The machine could filter and/or block messages depending on their content or even at random. Thus, the adversary could gather non-redundant information by sending more than one message per challenge.*

3.3 Specialized AnoA Game

In this section, we simplify the game-based definitions of AnoA by tailoring the game to the Tor protocol and by only considering statically corrupting passive adversaries. Our simplified game eases our analyses by abstracting away all details of Tor's UC formalization.

As in the more general definition of AnoA, we formalize anonymity as a challenge-response game, in which the adversary has to distinguish two scenarios. These scenarios are described by a simplified anonymity function α-SIMPLE that receives as inputs two senders S_0 and S_1, two recipients R_0 and R_1, and the challenge bit b. It then selects one sender and one recipient, based on the challenge bit and the considered anonymity notion.

```
Simplified AnoA Challenger for Tor Ch-simple($\alpha$-simple, $b$)
Initialize the Game
  Receive $N \subseteq \mathcal{E}$ and $L \subseteq \mathcal{E}^2$ as input.
  if $\alpha$-simple $= \alpha$-simple$_{\mathsf{SA}}$, let $N := N \setminus \mathcal{S}$.
  if $\alpha$-simple $= \alpha$-simple$_{\mathsf{RA}}$, let $N := N \setminus \mathcal{R}$.
  if $\alpha$-simple $= \alpha$-simple$_{\mathsf{REL}}$, let $N := N \setminus (\mathcal{S} \cup \mathcal{R})$.
  Set first = true
Upon message Input $= (\mathsf{S}, \mathsf{R})$
  SimulateTor($\mathsf{S}, \mathsf{R}$)
Upon message Challenge $= (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$
  if first = false then
    Output $\perp$.
  else
    Compute $(\mathsf{S}^*, \mathsf{R}^*) \leftarrow \alpha$-simple$(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$
    Set first := false.
    SimulateTor($\mathsf{S}^*, \mathsf{R}^*$)
  end if
Subroutine SimulateTor($\mathsf{S}, \mathsf{R}$)
  Let $\mathsf{C} \leftarrow$ ps$(\mathsf{S}, \mathsf{R})$ be a fresh Tor circuit.
  Send Observations := obs$[N, L](\mathsf{C})$ to A.
```

Figure 3.8: The simplified AnoA challenger for Tor against statically corrupting, passive adversaries, single-challenge messages and for one challenge.

Game-based anonymity definition We replace the complex challenger Ch with a significantly simpler challenger Ch-simple that only allows for one challenge that consists of exactly one message. Instead of simulating the UC formalization of Tor, it computes the adversary's observations via the observation function obs. For completeness we now describe the challenger in detail.

The AnoA challenger. The challenger Ch-simple is defined in Figure 3.8. As described above, it expects as inputs the anonymity notion α-simple and the challenge bit b. The challenger initially waits for a set $N \subseteq \mathcal{E}$ of compromised Tor entities and a set $L \subseteq \mathcal{E}^2$ of compromised connections between Tor entities. Upon receiving these sets, the challenger removes illegitimate corruption requests: $\mathcal{S}$ is removed from N for sender anonymity, $\mathcal{R}$ is removed from N for recipient anonymity, and both $\mathcal{S}$ and $\mathcal{R}$ are removed from N for relationship anonymity, which reflects the respective scenarios.

```
$\alpha\text{-SIMPLE}_{\mathsf{SA}}(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$ :
    output $(\mathsf{S}_b, \mathsf{R}_0)$

$\alpha\text{-SIMPLE}_{\mathsf{RA}}(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$ :
    output $(\mathsf{S}_0, \mathsf{R}_b)$

$\alpha\text{-SIMPLE}_{\mathsf{REL}}(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$ :
    $a \leftarrow^u \{0, 1\}$
    output $(\mathsf{S}_a, \mathsf{R}_{a \oplus b})$
```

Figure 3.9: The simplified (single-message) anonymity functions.

Then, it accepts two types of messages from the adversary: *challenge-messages*, denoted as `Challenge` $= (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$ in Figure 3.8, that trigger that the challenge message is sent, and *input-messages*, denoted as `Input` $= (\mathsf{S}, \mathsf{R})$ in Figure 3.8, that send additional messages m between senders S and recipients R :

- *Challenge-messages:* Upon receiving the (first) message `Challenge` $= (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1) \in \mathcal{S}^2 \times \mathcal{R}^2$, the challenger computes the anonymity notion α-SIMPLE on $(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$ and the challenge bit b and obtains a sender-recipient pair $(\mathsf{S}^*, \mathsf{R}^*) \in \{\mathsf{S}_0, \mathsf{S}_1\} \times \{\mathsf{R}_0, \mathsf{R}_1\}$. The challenger then very abstractly simulates the Tor protocol where S^* sends a message to R^*. We abbreviate this using the subroutine SIMULATETOR$(\mathsf{S}^*, m, \mathsf{R}^*, (\text{CH}, \Psi))$ in Figure 3.8. The challenger samples a new Tor circuit C from sender S^* to recipient R^* from the path selection distribution PS$(\mathsf{S}^*, \mathsf{R}^*)$ and sends the observations OBS that the adversary can make as OBS$[N, L](\mathsf{C})$ to A.

- *Input-messages:* Upon receiving a message `Input` $= (\mathsf{S}, \mathsf{R}) \in \mathcal{S} \times \mathcal{R}$, the challenger calls the subroutine SIMULATETOR(S, R), as described above, i.e., samples a fresh circuit and sends the observations to A.

Based on this simplified game we describe the reduction of anonymity of any statically corrupting passive adversary on Tor, analogously to Definition 2.2.1, as follows.

Definition 3.3.1 (Reduction of anonymity; advantage)**.** *Let α-SIMPLE be an anonymity function. Then, the* adversary's advantage *of an adversary* A *is at most δ, with $0 \leq \delta \leq 1$, if for all sufficiently large $\eta \in \mathbb{N}$, we have*

$$\begin{aligned} &\texttt{Pr}\left[0 = \langle \mathrm{A}(1^\eta) | \text{CH-SIMPLE}(\alpha\text{-SIMPLE}, 0)\rangle\right] \\ \leq\ & e^{\varepsilon}\ \texttt{Pr}\left[0 = \langle \mathrm{A}(1^\eta) | \text{CH-SIMPLE}(\alpha\text{-SIMPLE}, 1)\rangle\right] + \delta. \end{aligned}$$

We say that Tor exhibits a reduction of anonymity *of at most* (ε, δ) *(formally: Tor is* $(\alpha\text{-SIMPLE}, 1, \epsilon, \delta)$*-IND-CDP) against a class* $\mathcal{A}$ *of adversaries if the adversary's advantage of all probabilistic polynomial-time adversaries* $\mathrm{A} \in \mathcal{A}$ *is bounded by* ε *and* δ*.*

This definition captures an eavesdropping adversary that corrupts a fixed set of nodes and connections between nodes before it starts observing the network. In particular, the adversary cannot adaptively decide which nodes and connections to compromise.

Anonymity Function The anonymity guarantee we provide in AnoA is parametric in the function $\alpha\text{-SIMPLE} : \mathcal{S}^2 \times \mathcal{R}^2 \times \{0,1\} \to \mathcal{S} \times \mathcal{R}$, describing the anonymity notion. As for the more complex anonymity functions of AnoA, α-SIMPLE encodes the anonymity notion we analyze. We refer to Figure 3.9 for an overview over the three considered anonymity notions.

Adversary For our simplified game, we define a new adversary class $\mathrm{PnP}_{M,F_{\mathrm{OBS}},S_p}$ analogously to Definition 3.2.3, with a few minor, technical adaptations:

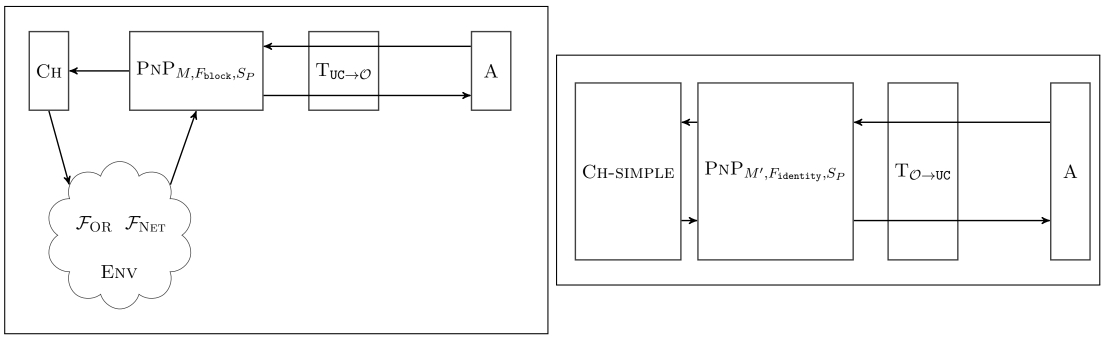
Definition 3.3.2 (Passive Observation Adversary with Transparent Challenges)**.** *Given any challenge Turing machine* M' *and a predicate on compromisation* P*, we construct a passive, statically corrupting Tor adversary class as a Plug'n'Play adversary class* $\mathrm{PnP}_{M,F_{\mathtt{block}},S_P}$ *as follows:*

- *Exactly as in Definition 3.2.3, the challenge machine* M *runs the input machine* M' *but enforces transparent challenges.*
- *The filter machine* F_{OBS} *relays all messages from* Ch *to* A*. Since the adversary cannot send messages to any protocol,* F_{OBS} *does not have to block such messages.*
- *The setup machine* S_P*, if initialized by the adversary with* $N \subseteq \mathcal{E}$ *and* $L \subseteq \mathcal{E}^2$*, checks, whether* $P(N, L) = \mathtt{true}$ *and only if so, allows this compromisation. To this end and in contrast to Definition 3.2.3,* S_P *then simply forwards* (N, L) *to* Ch-SIMPLE*.*

We will now show that the simplified AnoA game against a passive observation adversary with transparent challenges is equivalent to the more complex AnoA game using the UC formalization against a passive UC adversary with transparent challenges as in Definition 3.2.3.

3.3.1 Soundness of the Simplified Game

We show that quantifying anonymity based on the observations transmitted by our simplified challenger suffices for calculating the success of a passive, corruption-based



Ch
PnP$_{M,F_{\texttt{block}},S_P}$
T$_{\texttt{UC}\to\mathcal{O}}$
A
$\mathcal{F}_{\text{OR}}$ $\mathcal{F}_{\text{Net}}$
Env
Ch-simple
PnP$_{M',F_{\texttt{identity}},S_P}$
T$_{\mathcal{O}\to\texttt{UC}}$
A

Figure 3.10: The two generic translators $\mathrm{T}_{\mathsf{UC}\to\mathcal{O}}$ (left) and $\mathrm{T}_{\mathcal{O}\to\mathsf{UC}}$ (right) in the respective games in which we use them.

attacker, even when considering the significantly more complex UC definition of Tor. More precisely, we show that for every anonymity notion $\alpha \in \{\alpha\text{-SIMPLE}_{\mathsf{SA}}, \alpha\text{-SIMPLE}_{\mathsf{RA}}, \alpha\text{-SIMPLE}_{\mathsf{REL}}\}$, every adversary A (that can be expressed as in Definition 3.3.2) that wins the AnoA game with ε, δ can be directly translated to a corresponding adversary (expressed as in Definition 3.2.3) against the idealized version of Tor's UC protocol with an equal advantage (up to a negligible factor) and vice versa. We then combine the results from above and show that under the assumption of perfect traffic correlation, all AnoA guarantees result in equivalent guarantees for Tor's implementation (up to a negligible loss due to the cryptographic instantiation).

In the UC game, the adversary can typically only observe the handles of all messages transmitted to compromised Tor nodes or over observed links between Tor entities, and, in some cases the message itself. More precisely, the UC adversary can (passively) observe the following events and messages:

- All handles for messages that are sent to compromised Tor nodes $\mathsf{n} \in N$ and over compromised connections $(\mathsf{n}, \mathsf{n}') \in L$, which includes system messages like "extend circuit" and "circuit extended".

- The messages themselves, if the exit node or the recipient is compromised (the recipient can be a Tor node, for circuit handling messages) or the connection from exit to recipient is observed.

Since the answers of the recipient are computed by a stateless machine Rec and we require the UC adversary class (and, consequently, our observation adversary class) to have transparent challenges, the adversary already can compute all answers of recipients on its own by running Rec on the challenge messages.

Definition 3.3.3 (Generic Translation)**.** *We define two generic translators; a translator* $\mathrm{T}_{\mathsf{UC}\to\mathcal{O}}$ *that translates UC observations into abstract observations and a translator* $\mathrm{T}_{\mathcal{O}\to\mathsf{UC}}$ *that translates abstract observations into UC observations, as follows:*

- ***From observation tuple to UC*** ($\mathrm{T}_{\mathcal{O}\to\mathsf{UC}}$)***:*** *The machine first initializes the whole ideal functionality for Tor, as described in Section 3.2 for all Tor entities $\mathcal{E} \cup \{\mathsf{n}_{\mathtt{dummy}}\}$, where $\mathsf{n}_{\mathtt{dummy}} \notin \mathcal{E}$ is an additional protocol party that is not part of the regular Tor entities. We use $\mathsf{n}_{\mathtt{dummy}}$ to describe protocol parties that the adversary cannot observe.*

  Then, whenever the translator is queried with an observation $o = (\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3, \mathsf{n}_4, \mathsf{n}_5) \in \mathcal{O}$ and a message m, we replace all unobserved elements $\mathsf{n}_i = \bot$ in o with $\mathsf{n}_{\mathtt{dummy}}$. Subsequently, we ask the sender n_1 (who might be $\mathsf{n}_{\mathtt{dummy}}$) to create a fresh Tor circuit consisting of the nodes $\mathsf{n}_2, \mathsf{n}_3$ and n_4 and we then send the message m over this circuit to n_5. Note that any of the Tor nodes involved and/or the recipient may be $\mathsf{n}_{\mathtt{dummy}}$.[3] *We refer to Figure 3.12 for a formal description.*

- ***From UC transcripts to an observation tuple*** ($\mathrm{T}_{\mathsf{UC}\to\mathcal{O}}$)***:*** *Whenever the translator is queried with the transcript t of observations made by the compromised Tor entities and the observed links between them, $\mathrm{T}_{\mathsf{UC}\to\mathcal{O}}$ parses the transcript and counts the number of times it observes an entity. From this count we can infer the position of the entity on the circuit: as Tor creates circuits in a telescopic way, the number of messages sent for creating a circuit depends on the position of the entities; e.g., the guard node has to relay all messages, including the circuit construction messages and the corresponding responses, from the sender to the subsequent nodes, whereas the exit node only relays the message m itself to the recipient. We refer to Figure 3.11 for a formal description.*

Theorem 3.3.1. *For every challenge machine M', every predicate P, and every anonymity notion $\alpha \in \{\alpha_{\mathsf{SA}}, \alpha_{\mathsf{RA}}, \alpha_{\mathsf{REL}}\}$, the anonymity impact of the adversary class $\mathrm{PnP}_{M,F_{\mathtt{block}},S_P}$ (c.f. Definition 3.2.3) in the game with $\mathrm{Ch\text{-}SIMPLE}(\alpha\text{-}\mathrm{SIMPLE}, b)$ is equal to the anonymity impact of the corresponding adversary class $\mathrm{PnP}_{M,F_{\mathrm{OBS}},S_P}$ (c.f. Definition 3.3.2) on Π^{once} in the game with $\mathrm{Ch}(\alpha, 1, b)$, up to a negligible factor.*

Proof. Let M' be a challenge machine, P be a predicate and $\alpha \in \{\alpha_{\mathsf{SA}}, \alpha_{\mathsf{RA}}, \alpha_{\mathsf{REL}}\}$ be an anonymity function with α-SIMPLE as the corresponding simple anonymity notion. Further, let A be any PPT machine.

Simplified AnoA → AnoA with UC We show that for A, the game with the challenger Ch running the protocol Π^{once} is indistinguishable from the game with the challenger Ch-SIMPLE, if additionally we translate all observations to UC using the translator $\mathrm{T}_{\mathcal{O}\to\mathsf{UC}}$, as described in Figure 3.12.

The two games differ in their simulation of the Tor protocol and, (technically) in their handling of the compromisation. However, the compromisation is only technically

[3]For the purpose of this proof the same Tor node can used several times in the same circuit. We could circumvent this by introducing several dummy entities $\mathsf{n}_{\mathtt{dummy}_1}$ to $\mathsf{n}_{\mathtt{dummy}_5}$ for the individual positions. For simplicity we chose to introduce only one such entity.

```
$\mathrm{T}_{\mathsf{UC}\to\mathcal{O}}$ (t)
  Set $\textsc{counts}(x,y)$:=0 for every $(x,y) \in \mathcal{E}^2$
  Set $\mathsf{n}_1$:=$\perp$; $\mathsf{n}_2$:=$\perp$; $\mathsf{n}_3$:=$\perp$; $\mathsf{n}_4$:=$\perp$; $\mathsf{n}_5$:=$\perp$
  Parse the transcript $t$ and proceed as follows:
  upon parsing $(P, P_{\mathtt{next}}, \textsc{cid}, h)$:
    $\textsc{IncCount}(P, P_{\mathtt{next}})$
  upon parsing $(P, Q_1, \ldots, Q_u, m/h)$:
    $\textsc{IncCount}(P, Q_1); \textsc{IncCount}(Q_1, Q_2); \ldots; \textsc{IncCount}(Q_{u-1}, Q_u)$
  if there are no more messages in $t$ then
    for each $(P, P_{\mathtt{next}}) \in \mathcal{E}^2$ do
      switch $(\textsc{counts}(P, P_{\mathtt{next}}))$
      case 1:
        $\mathsf{n}_4$:=$P$; $\mathsf{n}_5$:=$P_{\mathtt{next}}$
      case 2:
        $\mathsf{n}_3$:=$P$; $\mathsf{n}_4$:=$P_{\mathtt{next}}$
      case 3:
        $\mathsf{n}_2$:=$P$; $\mathsf{n}_3$:=$P_{\mathtt{next}}$
      case 4:
        $\mathsf{n}_1$:=$P$; $\mathsf{n}_2$:=$P_{\mathtt{next}}$
    end for
  end if
  Output $(\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3, \mathsf{n}_4, \mathsf{n}_5)$
where we define IncCount as follows:
IncCount (counts,P,Q)
  if $\textsc{counts}(Q, P) = 0$ then $\textsc{counts}(P, Q)$+=1 // We only count the direction
  from the sender to the recipient.
```

Figure 3.11: Translation algorithm $\mathrm{T}_{\mathsf{UC}\to\mathcal{O}}$

different (the same Tor entities will be compromised and the same connections between them will be observed). Moreover, $\mathrm{T}_{\mathcal{O}\to\mathsf{UC}}$ will simulate the circuit creation and the sending of messages depending on the five-tuple received by Ch-simple. We describe the execution of the protocol in more detail now and argue why the adversary cannot distinguish the games.

In both cases, A initially sends messages `setup` to S_P that contain a set of Tor entities N and a set of connections between Tor entities L. In both games, the predicate P is evaluated on (N, L) and only if it is satisfied, the compromisation is allowed by S_P. In the UC game, S_P restricts N according to the anonymity notion and subsequently,

```
$\mathrm{T}_{\mathcal{O}\to\mathtt{UC}}(N, L)$
Initially:
    Initialize $\mathcal{F}_{\text{NET}}$ and initialize $\mathcal{F}_{\text{OR}}$ for every Tor entity $e \in \mathcal{E} \cup \{\mathsf{n}_{\mathtt{dummy}}\}$.
    Initialize a dummy adversary $\mathrm{A}_{\mathtt{UC}}(N, L)$ that stores and reflects all messages it receives. The adversary deletes or outputs its transcript upon request.
    $\mathrm{A}_{\mathtt{UC}}$ sends messages compromise to all Tor entities $\mathsf{n} \in N$ and messages observe to $\mathcal{F}_{\text{NET}}$ for all $(\mathsf{n}, \mathsf{n}') \in L$.
upon receiving an observation $o = (\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3, \mathsf{n}_4, \mathsf{n}_5)$ and a message $m$:
    Request that $\mathrm{A}_{\mathtt{UC}}$ deletes its transcript.
    Replace all entries $\mathsf{n}_i = \perp$ in $o$ with $\mathsf{n}_{\mathtt{dummy}}$.
    Send $(\mathtt{create\ circuit}, \text{PARTIES} = \langle \mathsf{n}_2, \mathsf{n}_3, \mathsf{n}_4 \rangle)$ to the $\mathcal{F}_{\text{OR}}$ party $\mathsf{n}_1$.
    After the circuit is created, retrieve the circuit ID CID from $\mathsf{n}_1$ and send $(\mathtt{send}, \mathsf{C} = \langle \mathsf{n}_1 \overset{\text{CID}_1}{\Leftrightarrow} \mathsf{n}_2 \Leftrightarrow \mathsf{n}_3 \Leftrightarrow \mathsf{n}_4 \rangle, \langle \mathtt{data}, (\mathsf{n}_5, m) \rangle)$ to $\mathsf{n}_1$
    Query $\mathrm{A}_{\mathtt{UC}}$ for its transcript $t$ and output the transcript $t$.
where we use the original ideal functionality $\mathcal{F}_{\text{OR}}$ that in turn uses the ideal network functionality $\mathcal{F}_{\text{NET}}$.
```

Figure 3.12: Translation algorithm $\mathrm{T}_{\mathcal{O}\to\mathtt{UC}}$

messages `compromise` and `observe` are sent to the respective Tor entities and to $\mathcal{F}_{\text{NET}}$. In the simplified game, the messages are sent to the challenger, that, in turn, restricts N in the same way.

All communication between A and M' is the same in both games. Whenever M' sends a message `Challenge` or a message `Input` to the challenger, the protocol is simulated. The games only differ in the observations made by the adversary. Note that these observations are not restricted by $F_{\mathtt{block}}$ or F_{OBS}.

- In the game with CH, the adversary observes handles sent by Tor entities to other Tor entities, depending on the compromised nodes and observed connections.
- In the game with CH-SIMPLE we translate an observation $o \in \mathcal{O}$ to a transcript using $\mathrm{T}_{\mathcal{O}\to\mathtt{UC}}$ and using the message sent by M' to the challenger. [4]

The execution of $\mathrm{T}_{\mathcal{O}\to\mathtt{UC}}$ on an observation o and a message m and the execution of Π^{once} (where the message m is sent) can only differ in the Tor entities that create a circuit, are parts of the circuit or pose as the recipient. In both cases a Tor circuit $\mathsf{C} = (\mathsf{S}_a, \mathsf{n}_{\mathsf{g}}, \mathsf{n}_{\mathsf{m}}, \mathsf{n}_{\mathsf{x}}, \mathsf{R}_b)$ is drawn from the same distribution $\text{PS}(\mathsf{S}_a, \mathsf{R}_b)$. All entities that are observed are the same in both games. Let $(\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3, \mathsf{n}_4, \mathsf{n}_5) := \text{OBS}[N, L](\mathsf{C})$ be the

[4] $\mathrm{T}_{\mathcal{O}\to\mathtt{UC}}$ receives this message from M' since we enforce transparent challenges.

observation in the simple AnoA game. For every n_i s.t. $\mathsf{n}_i = \bot$, the adversary neither has compromised the respective Tor entity, nor observes it via another Tor entity or via an observed connection. This, however, means that by definition of $\mathcal{F}_{\mathrm{OR}}$ and $\mathcal{F}_{\mathrm{NET}}$, the adversary does not receive any message containing this entity. Consequently, we can exchange the entity with the (always unobserved) dummy entity $\mathsf{n}_{\mathtt{dummy}}$ in our simulation and the adversary will receive exactly the same messages.

AnoA UC → Simplified AnoA We show that for A, the game with the challenger Ch-simple is indistinguishable from the game with the challenger Ch running the protocol Π^{once}, if additionally we translate all UC transcripts to observations using the translator $\mathrm{T}_{\mathtt{UC}\to\mathcal{O}}$, as described in Figure 3.11.

The two games differ in their simulation of the Tor protocol and, (as above) in their handling of the compromisation. $\mathrm{T}_{\mathtt{UC}\to\mathcal{O}}$ will construct an observation $o \in \mathcal{O}$ from a transcript of messages it receives by counting the number of messages sent by Tor entities. We describe the execution of the protocol in more detail now and argue why the adversary cannot distinguish the games.

The setup phase is equivalent, as argued for the previous case and results in the same Tor entities being considered compromised and the same connections between them being considered observed.

As argued above, All communication between A and M' is the same in both games. Whenever M' sends a message `Challenge` or a message `Input` to the challenger, the protocol is simulated. The games only differ in the observations made by the adversary. Note that these observations are not restricted by $F_{\mathtt{block}}$ or F_{OBS}.

- In the game with Ch, the adversary observes handles sent by Tor entities to other Tor entities, depending on the compromised nodes and observed connections.
- In the game with Ch-simple we translate an observation $o \in \mathcal{O}$ to a transcript using $\mathrm{T}_{\mathcal{O}\to\mathtt{UC}}$.

The executions of $\mathrm{T}_{\mathtt{UC}\to\mathcal{O}}$ on a transcript t scans the transcript for messages, learns the position of all Tor entities by counting the number of messages that are observed and then outputs an observation five-tuple. We now argue why the same five-tuple is sent to A in both cases.

In both cases a Tor circuit $\mathsf{C} = (\mathsf{S}_a, \mathsf{n}_\mathsf{g}, \mathsf{n}_\mathsf{m}, \mathsf{n}_\mathsf{x}, \mathsf{R}_b)$ is drawn from the same distribution $\textsc{ps}(\mathsf{S}_a, \mathsf{R}_b)$. All entities that are observed are the same in both games.

In the UC formalization of Tor, S_a creates a Tor circuit by first sending a message `create` to n_g. Upon receiving a response from n_g, it sends a second message `extend circuit` to n_g and n_g now sends a message `create` to n_m. After receiving a response, S_a now sends another message `extend circuit` to n_g, n_g relays it to n_m and n_m sends a message `create` to n_x. Finally, upon receiving a response `extended`, S_a sends a handle for the message m to n_g, n_g relays it to n_m, n_m relays it to n_x and n_x sends the message m to R_b.

We see that S_a sends 4 messages to $\mathsf{n_g}$; $\mathsf{n_g}$ sends 3 messages to $\mathsf{n_m}$; $\mathsf{n_m}$ sends 2 messages to $\mathsf{n_x}$ and $\mathsf{n_x}$ sends 1 message to R. Consequently, by counting the number of messages, we can compute the position of a Tor entity in the circuit. This works independently of whether a Tor entity is observed or whether a connection between Tor entities is observed, as long as connections are observed in an undirected manner. Note that we only count the messages sent in the direction from the sender to the recipient, not the (equal number of) replies sent in the opposite direction.

Since the transcript only contains the Tor entities that were observed by the Tor entities in N or via the connections in L, the observation created by $\mathrm{T}_{\mathtt{UC}\to\mathcal{O}}$ equals $\textsc{obs}[N, L](\mathsf{C})$, which is equal to the output of Ch-simple.

This concludes the proof. Since the adversary cannot distinguish the games (in either case), it will behave the same and its probability to output the correct challenge bit $b^* = b$ is the same up to a negligible advantage. $\square$

Chapter 4

Calculating Anonymity Guarantees

In this section, we show how to compute tight bounds on the anonymity provided by Tor in the presence of an adversary that can observe the communication at or between certain Tor entities. We begin by characterizing the impact that observations from Chapter 3 have on Tor's anonymity for the anonymity notions sender anonymity, recipient anonymity and relationship anonymity, based on the adversary's compromisation, the challenge senders and recipients, and the path selection algorithm (i.e., the distribution over Tor nodes). We then show that these impacts constitute tight bounds on the adversary's success for all *fixed* adversaries, i.e., for adversaries that are known in advance, both in AnoA and in a practical instantiation of Tor [15] (up to a negligible influence). For calculating these bounds, we are required to instantiate the probability distribution that describes Tor's path selection algorithm by an actual distribution, which we present for Tor's path selection algorithm as well as for several of its variants. Finally, we present a heuristic for efficiently calculating the impact of specific structural adversaries that compromise Tor nodes restricted only by a given *budget*.

4.1 Calculating Anonymity Guarantees for Observations

In Chapter 3 we defined the observations a passive adversary can make by eavesdropping on Tor communication. Note that in the simplified AnoA game of Section 3.3, which we have shown to be sufficient for our analysis, only one challenge exists and this challenge only consists of one message. Consequently, only one Tor circuit is created and any passive adversary can base its guess of the challenge bit b only on exactly this one observation of exactly one Tor circuit. In this section we calculate the probability for each observation to occur depending on the sender and recipient of a Tor circuit. For a given challenge message $\texttt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$ and for each of the anonymity notions of sender anonymity, recipient anonymity and relationship anonymity we calculate the difference in probability for each observation if the challenge bit is $b = 0$ in comparison to the challenge bit being $b = 1$.

Lemma 4.1.1. *In the specialized AnoA game from Definition 3.3.1, any adversary that is part of a Plug'n'Play adversary class with transparent challenges can only base its decision about the challenge bit b on exactly one message $\textsc{obs} \in \mathcal{O}$ that reflects the adversary's observations of one single Tor circuit, for every stateless path selection algorithm.*

Proof. Note that in the specialized AnoA game the adversary only sends one message `Challenge`. We show the Lemma by showing that the only information the adversary learns that in any way depends on the challenge bit b of the challenger is the message `Observations` that the challenger sends after receiving the (single) message `Challenge` from the adversary.

Note that for every Plug'n'Play adversary, the messages `Input`(S, R) are directly sent to the challenger. The adversary class by definition cannot base any decision on such messages. Similarly, the message `Observations` that the challenger sends to the adversary after receiving the message `Input`(S, R) cannot influence the behavior of the adversary class, as it is immediately sent to the adversary and not processed by any machine of the adversary class. Furthermore, as we assume $\textsc{ps}$ to be known (in general), the adversary can compute the observation itself by sampling a circuit C from the distribution $\textsc{ps}(\mathsf{S}, \mathsf{R})$ and by computing the resulting observation as in Definition 3.1.2.

For the messages `Input` generated by the adversary class itself, this argumentation does not apply. However, since the adversary class has *transparent challenges*, the adversary learns both challenge senders $\mathsf{S}_0, \mathsf{S}_1$ and both challenge recipients $\mathsf{R}_0, \mathsf{R}_1$. The observation for the challenge message only depends on these senders and recipients and on the challenge bit b. Thus, we can, for simplicity, assume that the adversary class does not generate messages `Input`, as they cannot help the adversary in guessing the bit, but also cannot harm the adversary, since the adversary class has transparent challenges and thus cannot use input messages to hide information about the challenges.

Under the above assumptions, the game consists only of a message `Challenge` to the challenger, followed by one message `Observations` and the adversary can only base its guess for the bit b on exactly this one message `Observations`. □

As the adversary can only base its decisions on one observation message, we continue by calculating the probability for each observation. We explain our intuition behind the statelessness of the path selection algorithm and the impact of this assumption on our analyses of Tor in Section 4.2.1.

4.1.1 Probabilities of Observations

For any given compromisation, consisting of a set of compromised Tor entities $N \subseteq \mathcal{E}$ and a set of compromised communications between Tor entities $L \subseteq \mathcal{E}^2$ and for any

```
$\textsc{Observation-Phase}(N, L, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$
  for each $z \in \{\mathsf{S}_0, \mathsf{S}_1\} \times \{\mathsf{R}_0, \mathsf{R}_1\} \times \mathcal{O}(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$ do
    $\textsc{store}[z] := 0$
  end for
  for each $(\mathsf{S}, \mathsf{R}) \in \{\mathsf{S}_0, \mathsf{S}_1\} \times \{\mathsf{R}_0, \mathsf{R}_1\}$ do
    $\textsc{ps}_{\mathsf{S},\mathsf{R}}$:=new $\textsc{ps}(\mathsf{S}, \mathsf{R})$
  end for
  for each $(\mathsf{S}, \mathsf{R}) \in \{\mathsf{S}_0, \mathsf{S}_1\} \times \{\mathsf{R}_0, \mathsf{R}_1\}, (\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}) \in \mathcal{N}^3$ do
    $\textsc{store}[\mathsf{S}, \mathsf{R}, \textsc{observed}(N, L, \mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R})]$+=$\textsc{ps}_{\mathsf{S},\mathsf{R}}(\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x})$
  end for
  return $\textsc{store}$
where we define

$\textsc{observed}(N, L, \mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R})$
  Initialize $i$:=$\perp$ for $i \in \{o_\mathsf{S}, o_\mathsf{G}, o_\mathsf{M}, o_\mathsf{X}, o_\mathsf{R}\}$
  if $\{\mathsf{S}, \mathsf{n_g}\} \cap N \neq \emptyset$ or $(\mathsf{S}, \mathsf{n_g}) \in L$ then $o_\mathsf{S}$:=$\mathsf{S}$
  if $\{\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}\} \cap N \neq \emptyset$ or $(\mathsf{S}, \mathsf{n_g}) \in L$ or $(\mathsf{n_g}, \mathsf{n_m}) \in L$ then $o_\mathsf{G}$:=$\mathsf{n_g}$
  if $\{\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}\} \cap N \neq \emptyset$ or $(\mathsf{n_g}, \mathsf{n_m}) \in L$ or $(\mathsf{n_m}, \mathsf{n_x}) \in L$ then $o_\mathsf{M}$:=$\mathsf{n_m}$
  if $\{\mathsf{n_m}, \mathsf{n_x}, \mathsf{R}\} \cap N \neq \emptyset$ or $(\mathsf{n_m}, \mathsf{n_x}) \in L$ or $(\mathsf{n_x}, \mathsf{R}) \in L$ then $o_\mathsf{X}$:=$\mathsf{n_x}$
  if $\{\mathsf{n_x}, \mathsf{R}\} \cap N \neq \emptyset$ or $(\mathsf{n_x}, \mathsf{R}) \in L$ then $o_\mathsf{R}$:=$\mathsf{R}$
  return $(o_\mathsf{S}, o_\mathsf{G}, o_\mathsf{M}, o_\mathsf{X}, o_\mathsf{R})$
```

Figure 4.1: Computation of the probability for every observation. Here (N,L) denotes compromised Tor entities and malicious infrastructure, S_0 and S_1 two senders, and R_0 and R_1 two recipients.

Tor circuit $\mathsf{C} = (\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R})$, we can compute the observation o the respective adversary makes, as $\textsc{obs}[N, L](\mathsf{C})$, as described in Definition 3.1.2. The probability that the circuit C is chosen depends on the path selection algorithm the sender uses, as well as, possibly, on the recipient. For now, we consider the path selection algorithm $\textsc{ps}$ as a distribution over Tor circuits that depends on the sender and the recipient. Consequently, we compute the probability of each observation as described in the algorithm $\textsc{Observation-Phase}$ in Figure 4.1, where $\textsc{ps}_{\mathsf{S},\mathsf{R}}(\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x})$ denotes the probability that $\textsc{ps}(\mathsf{S}, \mathsf{R})$ returns the circuit $\mathsf{C} = (\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R})$.

Based on the probability for every observation, we can compute the anonymity impact of every observation depending on the anonymity notion.

```
$\phi_\varepsilon(X, Y)$
  if $X > e^\varepsilon \cdot Y$ then
    return $X - e^\varepsilon \cdot Y$
  else
    return $0$
  end if
```

Figure 4.2: Impact of the difference of probabilities on our differential privacy style anonymity guarantee.

4.1.2 Distinguishing Anonymity Scenarios

For each of the three anonymity notions sender anonymity, recipient anonymity and relationship anonymity, the adversary needs to distinguish between the anonymity scenarios defined by the notion and the choice of senders and recipients. In each scenario, the adversary can make observations depending on the probability distribution over the circuits, as computed above.

Whenever the probability of an observation differs, say it is X for $b = 0$, but Y for $b = 1$, then the adversary can gain a small advantage, which we define as $\phi_\varepsilon(X, Y)$. The definition of $\phi_\varepsilon(X, Y)$ is inherently asymmetric, as we strive for an asymmetric anonymity definition (we bound the advantage in distinguishing between $b = 0$ and $b = 1$ by an inequation). We refer to Figure 4.2 for our definition of ϕ.

Lemma 4.1.2. *For every* PPT *distinguisher* A *that tries to distinguish the distributions X and Y over the same set U, the advantage of* A *is bound as follows.*

$$\begin{aligned}\texttt{Pr}\left[0 = \mathrm{A}(u) | u \leftarrow X\right] \leq e^\varepsilon\ &\texttt{Pr}\left[0 = \mathrm{A}(u) | u \leftarrow Y\right] \\ &+ \sum_{u \in U} \left(\phi_\varepsilon(\texttt{Pr}\left[x = u | x \leftarrow X\right], \texttt{Pr}\left[y = u | y \leftarrow Y\right])\right)\end{aligned}$$

Proof. The proof follows directly by the definition of ϕ_ε:

$$\begin{aligned}
&\texttt{Pr}\left[0 = \mathrm{A}(u) | u \leftarrow X\right] \\
&= \sum_{u \in U} \left(\texttt{Pr}\left[0 = \mathrm{A}(u)\right] \cdot \texttt{Pr}\left[u = x | x \leftarrow X\right]\right) \\
&\leq \sum_{u \in U} \left(\texttt{Pr}\left[0 = \mathrm{A}(u)\right] \cdot \begin{cases} e^\varepsilon \texttt{Pr}\left[u = y | y \leftarrow Y\right] \text{ if} & \texttt{Pr}\left[u = x | x \leftarrow X\right] \\ & \leq e^\varepsilon \texttt{Pr}\left[u = y | y \leftarrow Y\right] \\ \texttt{Pr}\left[u = x | x \leftarrow X\right] & \text{otherwise} \end{cases} \right) \\
&\leq \sum_{u \in U} \left(\texttt{Pr}\left[0 = \mathrm{A}(u)\right] \cdot \left(e^\varepsilon \texttt{Pr}\left[u = y | y \leftarrow Y\right] + \phi_\varepsilon(\texttt{Pr}\left[u = x | x \leftarrow X\right], \texttt{Pr}\left[u = y | y \leftarrow Y\right])\right)\right) \\
&\leq e^\varepsilon\ \texttt{Pr}\left[0 = \mathrm{A}(u) | u \leftarrow Y\right] + \sum_{u \in U} \left(\phi_\varepsilon(\texttt{Pr}\left[x = u | x \leftarrow X\right], \texttt{Pr}\left[y = u | y \leftarrow Y\right])\right)
\end{aligned}$$

$$\varepsilon\text{-}\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(N,L) := \sum_{o\in\mathcal{O}} \phi_\varepsilon\Big(\quad \mathtt{Pr}\left[o = \textsc{obs}[N,L](\mathsf{C}), \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_0,\mathsf{R}_0)\right], \quad \mathtt{Pr}\left[o = \textsc{obs}[N,L](\mathsf{C}), \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_1,\mathsf{R}_0)\right]\Big)$$

Figure 4.3: Calculation of $\varepsilon\text{-}\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(N,L)$ that defines the impact of any observation on sender anonymity.

$$\varepsilon\text{-}\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(N,L) := \sum_{o\in\mathcal{O}} \phi_\varepsilon\Big(\quad \mathtt{Pr}\left[o = \textsc{obs}[N,L](\mathsf{C}), \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_0,\mathsf{R}_0)\right], \quad \mathtt{Pr}\left[o = \textsc{obs}[N,L](\mathsf{C}), \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_0,\mathsf{R}_1)\right]\Big)$$

Figure 4.4: Calculation of $\varepsilon\text{-}\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(N,L)$ that defines the impact of any observation on recipient anonymity.

For the second to last inequation note that

$$\mathtt{Pr}\left[u = x | x \leftarrow X\right] = e^{\varepsilon}\mathtt{Pr}\left[u = y | y \leftarrow Y\right] + \mathtt{Pr}\left[u = x | x \leftarrow X\right] - e^{\varepsilon}\mathtt{Pr}\left[u = y | y \leftarrow Y\right].$$

□

As the calculation of the advantage depends on the anonymity scenarios that, in turn, depend on the anonymity notion, we discuss them for every notion individually.

Distinguishing Sender Anonymity Scenarios For sender anonymity, the scenarios defined by the challenge message $\mathtt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, m)$ are that either S_0 or S_1 send the message m to R_0 (we refer to Section 1.2.2 for an intuition on these scenarios).

The observations an adversary can make only depend on the probability distribution over Tor circuits for each scenario and the scenarios only differ in the sender: a circuit is drawn either from $\textsc{ps}(\mathsf{S}_0, \mathsf{R}_0)$ or from $\textsc{ps}(\mathsf{S}_1, \mathsf{R}_0)$. We thus calculate the impact of an adversary that compromises the Tor entities N and the communications between Tor entities L on sender anonymity $\varepsilon\text{-}\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(N,L)$ as defined in Figure 4.3.

Distinguishing Recipient Anonymity Scenarios For recipient anonymity, the scenarios defined by the challenge message $\mathtt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, m)$ are that S_0

$$\begin{aligned}\varepsilon\texttt{-impact}^{\mathcal{O}}_{\mathsf{REL}}(N,L) := \sum_{o\in\mathcal{O}} \phi_\varepsilon\Big(&\big(\texttt{Pr}\left[o = \textsc{obs}[N,L](\mathsf{C}), \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_0,\mathsf{R}_0)\right] \\ &+ \texttt{Pr}\left[o = \textsc{obs}[N,L](\mathsf{C}), \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_1,\mathsf{R}_1)\right]\big)/2, \\ &\big(\texttt{Pr}\left[o = \textsc{obs}[N,L](\mathsf{C}), \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_0,\mathsf{R}_1)\right] \\ &+ \texttt{Pr}\left[o = \textsc{obs}[N,L](\mathsf{C}), \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_1,\mathsf{R}_0)\right]\big)/2\Big)\end{aligned}$$

Figure 4.5: Calculation of $\varepsilon\texttt{-impact}^{\mathcal{O}}_{\mathsf{REL}}(N,L)$ that defines the impact of any observation on relationship anonymity.

sends the message m either to R_0 or to R_1 (we refer to Section 1.2.2 for an intuition on these scenarios).

Analogously to sender anonymity, the observations an adversary can make for the two scenarios differ in the probability distributions over Tor circuits, which only differ depending on the recipient: a circuit is drawn either from $\textsc{ps}(\mathsf{S}_0,\mathsf{R}_0)$ or from $\textsc{ps}(\mathsf{S}_0,\mathsf{R}_1)$. We thus calculate the impact of an adversary that compromises the Tor entities N and the communications between Tor entities L on recipient anonymity $\varepsilon\texttt{-impact}^{\mathcal{O}}_{\mathsf{RA}}(N,L)$ as defined in Figure 4.4.

Distinguishing Relationship Anonymity Scenarios Relationship anonymity is slightly more complex than sender anonymity and recipient anonymity. Recall that for relationship anonymity there are the following four anonymity scenarios for every challenge message $\texttt{Challenge} = (\mathsf{S}_0,\mathsf{S}_1,\mathsf{R}_0,\mathsf{R}_1)$: any of the two senders sends the message m to any of the two recipients and the adversary has to distinguish S_x communicating with R_x for $x \in \{0,1\}$ from S_x communicating with R_{1-x} for $x \in \{0,1\}$.

The observations the adversary can make for the anonymity scenarios differ in the probability distributions over Tor circuits. However, the adversary does not a priori know the choice of x, which we have to consider for modeling the impact. If the adversary observes R_0 as the recipient of a message, it does not know whether $x = 0$ and $b = 0$ or whether $x = 1$ and $b = 1$. We thus compare the probability for an observation depending on the challenge bit, not on the scenario itself by combining the probability distributions for $\textsc{ps}(\mathsf{S}_0,\mathsf{R}_0)$ with $\textsc{ps}(\mathsf{S}_1,\mathsf{R}_1)$ and $\textsc{ps}(\mathsf{S}_0,\mathsf{R}_1)$ with $\textsc{ps}(\mathsf{S}_1,\mathsf{R}_0)$ and calculate the impact of an adversary that compromises the Tor entities N and the communications between Tor entities L on recipient anonymity $\varepsilon\texttt{-impact}^{\mathcal{O}}_{\mathsf{REL}}(N,L)$ as defined in Figure 4.5.

Intuitions for the Observation Impact Alternatively to the observation-based quantification presented here one can discuss the reduction of anonymity in terms of *distinguishing observations* and *evidence-building observations*. An observation is distinguishing if it clearly helps in distinguishing the scenarios, e.g., a compromised guard node can clearly distinguish two senders and thus break sender anonymity. Evidence-building observations, however, do not lead to an immediate deanonymization, but present some slight evidence that can be used for distinguishing the scenarios with higher probability, e.g., a compromised middle node could observe an exit node that is more likely to be used for communicating with one recipient than for communicating with another recipient.

Our observation-based quantification already suffices for quantifying both types of observations. Note that the set of possible observations for a circuit also includes the possible senders and recipients. If for sender anonymity, where either Alice or Bob send a message, a guard node is compromised, the adversary can for one of the scenarios observe that Alice sends a message, resulting in some observation $o = (\texttt{Alice}, \mathsf{n}_{\mathtt{g}}, \_, \_, \_)$. This observation, however, is impossible if actually Bob sends a message in the other scenario. Consequently, the difference in probabilities immediately characterizes the probability that the guard node is compromised, thus reflecting a distinguishing observation.

4.1.3 A Sound and Precise Calculation of the Reduction of Anonymity

So far we have presented a way for calculating the probabilities of observations depending on the probability distribution over Tor circuits defined Tor's path selection algorithm. Moreover we have discussed the impact of these observations on the anonymity scenarios defined by an adversarial challenge, the anonymity notion and the challenge bit.

We now combine these insights into an algorithm that quantifies the reduction of anonymity of one specific challenge message $\texttt{Challenge}(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$ depending on a set of compromised Tor entities N and a set of compromised communications between Tor entities L and then show that this algorithm is sound, and, under the assumption that traffic correlation is perfect, tight. Our algorithm calculates the reduction of anonymity for all three anonymity notions at once. To this end, it starts by defining sets of compromised Tor entities for each of the anonymity scenarios, strictly following the setup of the (simplified) AnoA game: For sender anonymity, the set N_{SA} may not contain any sender or recipient (in our case we simply remove S_0 and S_1); for recipient anonymity, the set N_{RA} may not contain any recipient (in our case we simply remove R_0 and R_1); for relationship anonymity, the set N_{REL} may neither contain senders nor recipients (in our case we simply remove all four).

Subsequently, for every anonymity notion we calculate the probability of every observation, which we call the *observation phase*, as presented in Figure 4.1, based on the challenge senders and recipients and on the respective set N_X for $X \in \{\mathsf{SA}, \mathsf{RA}, \mathsf{REL}\}$. Then, in the so called *deduction phase*, our algorithm calculates the impact of all ob-

```
COMPUTEDELTA($\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, N, L, \varepsilon$)
  $N_{\mathsf{SA}} := N \setminus \{\mathsf{S}_0, \mathsf{S}_1\}$; $N_{\mathsf{RA}} := N \setminus \{\mathsf{R}_0, \mathsf{R}_1\}$; $N_{\mathsf{REL}} := N \setminus \{\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1\}$.
  for $X \in \{\mathsf{SA}, \mathsf{RA}, \mathsf{REL}\}$ do
    STORE$_X$ := OBSERVATION-PHASE($N_X, L, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1$)
    $\delta_X$ := DEDUCTION-PHASE(STORE, $X, \varepsilon, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1$)
  end for
  return ($\delta_{\mathsf{SA}}, \delta_{\mathsf{RA}}, \delta_{\mathsf{REL}}$)

DEDUCTION-PHASE(STORE, $X, \varepsilon, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1$)
  $\delta_{\mathsf{SA}}, \delta_{\mathsf{RA}}, \delta_{\mathsf{REL}} := 0$
  for each $o \in \mathcal{O}(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$ do
    $\delta_{\mathsf{SA}}$ += $\phi_\varepsilon$(STORE[$\mathsf{S}_0, \mathsf{R}_0, o$], STORE[$\mathsf{S}_1, \mathsf{R}_0, o$])
    $\delta_{\mathsf{RA}}$ += $\phi_\varepsilon$(STORE[$\mathsf{S}_0, \mathsf{R}_0, o$], STORE[$\mathsf{S}_0, \mathsf{R}_1, o$])
    $r_1$ := (STORE[$\mathsf{S}_0, \mathsf{R}_0, o$] + STORE[$\mathsf{S}_1, \mathsf{R}_1, o$])/2
    $r_2$ := (STORE[$\mathsf{S}_0, \mathsf{R}_1, o$] + STORE[$\mathsf{S}_1, \mathsf{R}_0, o$])/2
    $\delta_{\mathsf{REL}}$ += $\phi_\varepsilon(r_1, r_2)$
  end for
  return $\delta_X$

$\phi_\varepsilon(X, Y)$
  if $X > e^\varepsilon \cdot Y$ then
    return $X - e^\varepsilon Y$
  else
    return 0
  end if
```

Figure 4.6: Computation of reduction of anonymity. Here (N,L) denotes compromised Tor entities and malicious infrastructure, S_0 and S_1 two senders, and R_0 and R_1 two recipients.

servations on anonymity and computes a value for δ depending on a given value for ε as discussed in the previous subsection. We refer to Figure 4.6 for a description of our algorithm.

We now show that for all anonymity notions α_X with $X \in \{\mathsf{SA}, \mathsf{RA}, \mathsf{REL}\}$, the values for δ_X computed by our algorithm constitutes a tight guarantee for the anonymity notion α_X against an adversary that compromises exactly N and L and that sends exactly one challenge with the Tor entities $\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1$ as in the calculation.

Theorem 4.1.1. *Let $N \subseteq \mathcal{E}$ and $L \subseteq \mathcal{E}^2$ denote the sets of observation points, let $\mathsf{S}_0, \mathsf{S}_1 \in \mathcal{S}$ be two senders, $\mathsf{R}_0, \mathsf{R}_1 \in \mathcal{R}$ be two recipients, and let α_X for $X \in$*

$\{\mathsf{SA}, \mathsf{RA}, \mathsf{REL}\}$ *be an anonymity notion. And let* $\mathcal{A}^{N,L,\alpha_X}_{\mathsf{S}_0,\mathsf{S}_1,\mathsf{R}_0,\mathsf{R}_1}$ *be the adversary class* $\text{PNP}_{M,F_{\texttt{block}},S_P}$ *with* M *and* P *defined as:*

M: *Whenever invoked with a message* $\texttt{Challenge} = (\mathsf{S}_x, \mathsf{S}_y, \mathsf{R}_x, \mathsf{R}_y)$*, send the message* $\texttt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1)$ *to the challenger.*

P: *The predicate* P *evaluated on* (N', L') *depends on the anonymity notion* α_X*: If* $\alpha_X = \alpha_{\mathsf{SA}}$*, return true iff* $N' = N \setminus \mathcal{S}$ *and* $L' = L$*; if* $\alpha_X = \alpha_{\mathsf{RA}}$*, return true iff* $N' = N \setminus \mathcal{R}$ *and* $L' = L$*; if* $\alpha_X = \alpha_{\mathsf{REL}}$*, return true iff* $N' = N \setminus (\mathcal{S} \cup \mathcal{R})$ *and* $L' = L$*.*

Then Tor is $(\alpha_X, 1, \varepsilon, \delta)$*-IND-CDP) against* $\mathcal{A}^{N,L,\alpha_X}_{\mathsf{S}_0,\mathsf{S}_1,\mathsf{R}_0,\mathsf{R}_1}$*, for every* $\varepsilon \geq 0$*, and for* $\delta = \delta_X$*, as computed by* COMPUTEDELTA *(defined in Figure 4.6). Moreover, for every value* $\varepsilon \geq 0$*, there is an adversary* $\mathrm{A} \in \mathcal{A}^{N,L,\alpha_X}_{\mathsf{S}_0,\mathsf{S}_1,\mathsf{R}_0,\mathsf{R}_1}$ *that has advantage* $\delta = \delta_X$ *against Tor, if we assume perfect traffic correlation.*

Proof. **Soundness.** Let $\alpha_X \in \{\alpha_{\mathsf{SA}}, \alpha_{\mathsf{RA}}, \alpha_{\mathsf{REL}}\}$ be an anonymity notion and $\mathrm{A} \in \mathcal{A}^{N,L,\alpha_X}_{\mathsf{S}_0,\mathsf{S}_1,\mathsf{R}_0,\mathsf{R}_1}$ be an adversary from the considered class and let C be the circuit created by the challenger via $\mathsf{C} \leftarrow \text{PS}(\mathsf{S}^*, \mathsf{R}^*)$, where $\mathsf{S}^*, \mathsf{R}^*$ were computed by α_X. By Lemma 4.1.1 the adversary can base its decision only on one observation, namely $\text{OBS}[N, L](\mathsf{C})$ and thus, this decision directly depends on the probability to make the respective observation. Consequently, by Lemma 4.1.2, the advantage of A in the AnoA game is limited by ε and $\delta = \delta_X$, where δ_X was calculated by $\text{COMPUTEDELTA}(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, N, L, \varepsilon)$.
Tightness. We first discuss the tightness for sender anonymity. The proof is analogous for recipient anonymity and relationship anonymity. For the tightness we construct the following adversary B for sender anonymity.

- B initializes and sends a message $\texttt{Challenge}(\_, \_, \_, \_)$ to its adversary class, such that the challenge message is triggered.
- Moreover, B bases its decisions on a subset $\mathcal{O}_{\mathsf{SA}}$ of all possible observations $\mathcal{O}$, where $\mathcal{O}_{\mathsf{SA}} \subseteq \mathcal{O}$ contains all observations that are (sufficiently) more likely if S_0 is the sender (in comparison to S_1). More formally:

  $$\mathcal{O}_{\mathsf{SA}} := \{o \in \mathcal{O} \text{ if } \mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_0,o} \geq e^{\varepsilon} \cdot \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o}\},$$

  where

  $$\mathsf{OP}_{\mathsf{S},\mathsf{R},o} := \texttt{Pr}\left[o = \text{OBS}[N, L](c) | c \leftarrow \text{PS}(\mathsf{S}, \mathsf{R})\right]$$
- Upon (after sending the challenge) receiving a message $\texttt{Observations}(o)$ from CH-SIMPLE, B outputs $b^* := 0$ if and only if $o \in \mathcal{O}_{\mathsf{SA}}$ and $b^* := 1$ otherwise.

We now calculate the advantage of B in the game by comparing the probability that it outputs $b^* = 0$ if $b = 0$ with the probability that it outputs $b^* = 0$ if $b = 1$. We first show the Theorem for $\varepsilon = 0$, as this presents an intuition for our approach and is slightly less complex than the case for $\varepsilon > 0$.

$$\begin{aligned}
&\mathtt{Pr}\left[0 = \langle \mathrm{B}(1^\eta) | \textsc{Ch-simple}(\alpha\text{-}\textsc{simple}_{\mathsf{SA}}, 0)\rangle\right] \\
&-\mathtt{Pr}\left[0 = \langle \mathrm{B}(1^\eta) | \textsc{Ch-simple}(\alpha\text{-}\textsc{simple}_{\mathsf{SA}}, 1)\rangle\right] \\
&= \sum_{o \in \mathcal{O}_{\mathsf{SA}}} \mathtt{Pr}\left[o = \textsc{obs}[N_{\mathsf{SA}}, L](\mathsf{C}) | \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_0, \mathsf{R}_0)\right] \\
&- \sum_{o \in \mathcal{O}_{\mathsf{SA}}} \mathtt{Pr}\left[o = \textsc{obs}[N_{\mathsf{SA}}, L](\mathsf{C}) | \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_1, \mathsf{R}_0)\right] \\
&= \sum_{o \in \mathcal{O}_{\mathsf{SA}}} (\mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_0,o} - \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o}) \\
&= \sum_{o \in \mathcal{O}} (\phi_\varepsilon(\mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_0,o}, \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o})) \\
&= \varepsilon\text{-}\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(N_{\mathsf{SA}}, L) = \delta_{\mathsf{SA}} \text{ as computed by } \textsc{ComputeDelta}(\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, N, L, \varepsilon)
\end{aligned}$$

For $\varepsilon = 0$, the calculation from above directly yields our guarantee. For $\varepsilon > 0$ we show the theorem via contradiction. Assume that for a value ε there would be a δ' with $\delta' < \delta_{\mathsf{SA}}$, s.t., B cannot distinguish the scenarios better than with ε and δ' and where $\delta_{\mathsf{SA}} = \varepsilon\text{-}\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(N_{\mathsf{SA}}, L)$. For brevity we write $\textsc{Ch}$ instead of $\textsc{Ch-simple}$ and α_{SA} instead of $\alpha\text{-}\textsc{simple}_{\mathsf{SA}}$ in the following calculation.

$$\begin{aligned}
& & \mathtt{Pr}\left[0 = \langle \mathrm{B}(1^\eta) | \textsc{Ch}(\alpha_{\mathsf{SA}}, 0)\rangle\right] &\le e^\varepsilon \mathtt{Pr}\left[0 = \langle \mathrm{B}(1^\eta) | \textsc{Ch}(\alpha_{\mathsf{SA}}, 1)\rangle\right] + \delta' \\
&\Leftrightarrow & \sum_{o \in \mathcal{O}_{\mathsf{SA}}} \mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_0,o} &\le \sum_{o \in \mathcal{O}_{\mathsf{SA}}} (e^\varepsilon \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o}) + \delta' \\
&\Leftrightarrow & \sum_{o \in \mathcal{O}_{\mathsf{SA}}} (\mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_0,o} + e^\varepsilon \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o} - e^\varepsilon \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o}) &\le \sum_{o \in \mathcal{O}_{\mathsf{SA}}} (e^\varepsilon \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o}) + \delta' \\
&\Leftrightarrow & \sum_{o \in \mathcal{O}_{\mathsf{SA}}} (e^\varepsilon \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o} + \phi_\varepsilon(\mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_0,o}, \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o})) &\le \sum_{o \in \mathcal{O}_{\mathsf{SA}}} (e^\varepsilon \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o}) + \delta' \\
&\Leftrightarrow & \sum_{o \in \mathcal{O}_{\mathsf{SA}}} \phi_\varepsilon(\mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_0,o}, \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o}) &\le \delta' \\
&\Leftrightarrow & \delta_{\mathsf{SA}} &\le \delta'
\end{aligned}$$

This contradicts our assumption that $\delta' < \delta_{\mathsf{SA}}$ and thus concludes the proof.

We omit the proofs for recipient anonymity and relationship anonymity, as they follow completely analogously with the following two machines B_{RA} and $\mathrm{B}_{\mathsf{REL}}$.

- B_{RA} is defined exactly as B with the only difference that we use the set $\mathcal{O}_{\mathsf{RA}}$ instead of $\mathcal{O}_{\mathsf{SA}}$, where

$$\mathcal{O}_{\mathsf{RA}} := \{o \in \mathcal{O} \text{ if } \mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_0,o} \ge e^\varepsilon \cdot \mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_1,o}\}$$

- We analogously define the set $\mathcal{O}_{\mathsf{REL}}$ for $\mathrm{B}_{\mathsf{REL}}$, but combine the observations of two scenarios:

$$\begin{aligned}\mathcal{O}_{\mathsf{REL}} := \ & \{o \in \mathcal{O} \text{ if } (\mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_0,o} + \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_1,o})/2 \\ & \geq e^{\varepsilon} \cdot (\mathsf{OP}_{\mathsf{S}_0,\mathsf{R}_1,o} + \mathsf{OP}_{\mathsf{S}_1,\mathsf{R}_0,o})/2\}\end{aligned}$$

□

Finally, for calculating the impact on Tor, we need to instantiate the probability distribution $\mathrm{PS}_{\mathsf{S},\mathsf{R}}$ to accurately match the distribution over Tor circuits that the Tor client of the sender S chooses to communicate with R. To this end, we describe Tor's path selection algorithm in the next section.

4.2 Tor Path Selection Algorithms

Tor's anonymity guarantees inherently depend on its path selection algorithm, as this algorithm determines the probability by which nodes are chosen for a circuit.

In this section we introduce Tor's current path selection algorithm and discuss the technical restrictions it places on Tor nodes. Moreover, we introduce a few relevant variants of Tor's path selection algorithm: (a) Uniform-weights, a performance insensitive variant of Tor's path selection algorithm, (b) SelekTOR, a variant of Tor's path selection algorithm that restricts the exit node to nodes from a specific country, (c) LASTor, an approach to improve anonymity against malicious network infrastructure, and (d) DistribuTor, our novel variant of Tor's path selection algorithm that intents to improve anonymity against very bandwidth-extensive Tor nodes.

4.2.1 Tor

Tor's path selection algorithm does *not* select Tor nodes (uniformly) at random. To improve the performance, Tor's current path selection algorithm makes a weighted random choice over all nodes that support the user's connections and preferences, and bases the weights on information that is retrieved from a periodically published *server descriptor* and an hourly published *consensus document*. These documents are generated and maintained by a small set of semi-trusted directory authorities, and contain up-to-date information about each node.

In a server descriptor, a Tor node publishes its bandwidth, the ports it would allow as an exit node, its so-called *family* (used for distributing trust), its uptime, its operating system, and its version of Tor. In order to prevent malicious Tor nodes from equivocating (i.e., sending different information to different users), the nodes are required to periodically upload (every 12 to 18 hours) their current server descriptor to all directory authorities.

The consensus document is computed hourly by the directory authorities, and it contains for each node information such as the node's availability, its entrusted bandwidth, a pointer to the up-to-date server descriptor, and whether this node should be used as an exit node and/or an entry node. Moreover, the consensus document contains entry, middle, and exit scaling factors for every node in order to balance the bandwidth. This scaling is necessary since there are fewer nodes that are marked as exit nodes (≈1000 in October 2015) than as entry (≈1800 in October 2015), from an even larger set of Tor nodes (≈ 6500 in October 2015), most of which, however, can only be used as middle nodes.

The PSTor algorithm computes the weight of a node based on the retrieved node's entrusted bandwidth. Since a circuit establishment is expensive, the path selection tries to include as many of the requested ports into one circuit as possible. Given a list of requested ports by the user, PSTor determines the maximal set of ports that is supported by any exit node, and then excludes all nodes that do not support this maximal set of ports and that are not marked as exit nodes. Then, the algorithm assigns a weight to every remaining n by dividing its entrusted bandwidth $\mathsf{n}.bw$ by the sum of the entrusted bandwidths s of all not excluded nodes and multiplies this with the corresponding exit scaling factor $scEx(\mathsf{n})$ from the consensus document: $\mathsf{n}.bw/s * scEx(\mathsf{n})$. Finally, the algorithm performs a weighted random choice over these nodes.

As Tor is built upon the principle of distributing trust, the path selection excludes circuits with nodes that are *related*, i.e., that are in the same /16 subnet and nodes that are in each other's family. After having chosen the exit node, the path selection chooses an entry node in two steps: first, the algorithm excludes all nodes that are related to the exit node and all nodes that are not marked as entry nodes in the current consensus; second, the algorithm computes the weight of each of the remaining nodes by dividing their entrusted bandwidth by the sum of all not excluded nodes and performs a weighted random choice over these nodes. For the middle node the path selection proceeds as for the entry nodes except that middle nodes do not require specific tags. However, all relatives of both the exit node and the entry node are excluded.

This path selection algorithm adapts to the preferences of the user, who can, e.g., decide to only use nodes that have the 'stable' tag or to build circuits that only use 'fast' nodes. Tor's path selection algorithm also offers a configuration for including non-valid entry or exit nodes as well as entry nodes that are not considered to be entry guards. We refer to Tor's specification [41] for a more detailed description.

4.2.2 DistribuTor

The bandwidth of Tor nodes is not uniformly distributed as Tor tries improve its performance by selecting nodes depending on their bandwidth. As a result, a node with twice the bandwidth is twice as likely to be used for a circuit. The real-life bandwidth distribution, however, contains nodes that are several hundred times as likely as other nodes. Consequently, a small number of nodes with a very high bandwidth is used in a large percentage of circuits. If these nodes get compromised or similar new nodes are added to the network by an adversary, this adversary can deanonymize many connections. Thus, the current path selection of Tor produces obvious targets such that an attacker that compromises these points can deanonymize a significant part of the network.

Novel loss-less path selection: Re-balancing the workload To reduce the risk posed by such high bandwidth nodes we propose PSDistribuTor, a path selection algorithm that distributes the trust amongst exit and entry nodes as evenly as possible. We observe that the exit bandwidth inherently is a bottleneck as only few nodes wish to be used as exit nodes. Hence, we first focus on the exit nodes.

- **Distributing the bandwidth for exit nodes:** We compute the exit bandwidth that Tor supports at a given moment by summing over the bandwidth of all possible exit nodes and weighting them according to their tags and the weights from the consensus. We then compute how evenly we can distribute the bandwidth by using small exit nodes solely as exit nodes and restricting the usage of the largest exit nodes accordingly. In this process we make sure that the previous exit bandwidth is still provided by our improved path selection algorithm.
- **Distributing the bandwidth for entry nodes:** After the weight of nodes for their position as an exit node has been set we compute the weights for entry nodes. We proceed just as before, by trying to preserve the entry bandwidth and still distributing the trust in entry nodes as widely as possible.

Anonymity improvement As we put a bound for the maximal weight of exit and entry nodes, we use their remaining bandwidth by increasing their weight to be used as middle node, as this position is considered least critical. The details of our redistribution can be found in Figure 4.7. In Section 5.1 we present our analysis computed on the real consensus data of Tor and evaluate PSDistribuTor against Tor's path selection algorithm.

Naturally it would be possible to sacrifice performance of the Tor network for a much stronger improvement in anonymity by reducing the targeted total bandwidth. In an extreme case one could weight all nodes uniformly, which would allow much stronger anonymity guarantees against a small number of colluding nodes.

Note that we did not consider the case that the entry bandwidth poses a bottleneck for Tor. In this case, one should change the order in which these calculations are made.

4.2.3 Other Variants

Uniform Node Selection Many existing works abstract Tor's actual path selection as a uniform path selection algorithm. We analyze not a strictly uniform routing

```
n.$exitBW$
1: if n can be used as exit then
2:    if n.$bw < maxExitBW$ then
3:       return n.$bw$
4:    else return $maxExitBW$
5:    end if
6: else return 0
7: end if

n.$entryBW$
1: if n can be used as entry then
2:    if n can be used as exit then
3:       if n.$bw < maxExitBW$ then
4:          return 0
5:       else
6:          if n.$bw - maxExitBW < maxEntryBW$ then
7:             return n.$bw - maxExitBW$
8:          else return $maxEntryBW$
9:          end if
10:      end if
11:   else return $\min(maxEntryBW, \mathsf{n}.bw)$
12:   end if
13: else return 0
14: end if

n.$middleBW$
1: $bw :=$ n.$bw$
2: if n can be used as exit then $bw := bw - maxExitBW$
3: if n can be used as entry then $bw := bw - maxEntryBW$
4: if $bw > 0$ then return $bw$
5: else return 0
```

Figure 4.7: PSDistribuTor: Our redistribution of the bandwidths

strategy, but a *parameter respecting* uniform routing strategy. In this variant of Tor, nodes can only be chosen as guard nodes, middle nodes or exit nodes if their parameters support that and the order in which they are chosen is equal to the order in which Tor selects these nodes. Related nodes (nodes in the same family or the same /16 subnet) cannot be chosen for the same circuit. However, all nodes are chosen with the same weight, ignoring the performance improvement strategies of Tor.

SelekTOR SelekTOR [89] restricts the Tor client to always select an exit node from a specific country, e.g., in order to bypass geo-restrictions of websites and services. SelekTOR only differs from TorPS in that the weights of all Tor exit nodes outside the considered country are set to zero. In our evaluation, we consider a SelekTOR configuration with exit nodes in the US.

LASTor LASTor [1] groups Tor nodes together into so-called clusters based on their physical location (latitude and longitude), which it infers by their IP address via GeoIP. LASTor first selects a guard cluster, a middle cluster, and an exit cluster and weights the guard cluster (and exit cluster) inverse to the distance between them and the sender (or the recipient, respectively), thereby reducing the expected physical distance. After selecting clusters, LASTor selects a node from each cluster uniformly at random.

4.2.4 Assumptions on Tor Path Selection Algorithms

For showing the soundness and precision of our guarantees we require some assumptions on the path selection algorithms. These assumptions are quite natural, although they may present a limitation for some analyses.

We assume that the path selection algorithm is a stateless probability distribution $\textsc{ps}$ parametric in a sender $\mathsf{S} \in \mathcal{S}$ and a recipient $\mathsf{R} \in \mathcal{R}$ over Tor circuits $\mathsf{C} \in \mathsf{Circuits}$. Moreover, all Tor circuits $\mathsf{C} = (\mathsf{S}', \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}')$ for which the path selection algorithm $\textsc{ps}(\mathsf{S}, \mathsf{R})$ has a non-zero probability satisfy the following properties: $\mathsf{S} = \mathsf{S}'$, $\mathsf{R} = \mathsf{R}'$, and all Tor nodes are pairwise distinct, i.e., $\mathsf{n_g} \neq \mathsf{n_m}$, $\mathsf{n_g} \neq \mathsf{n_x}$, $\mathsf{n_m} \neq \mathsf{n_x}$.

Remark 4.2.1 (Statelessness of the Path Selection Algorithm)**.** *We assume the path selection algorithm(s) we consider to be stateless. This follows our intuition that no state from any one input session should be transferred to a challenge session. and it holds true for many variants of Tor's path selection algorithm. However, the concept of entry guards, where a user selects one guard node that will be the guard node in every circuit. Our analysis, consequently, considers the user to change his guards for the challenge circuit.*

Alternatively, for analyzing guard nodes, we could rephrase the proofs in this section and consider an adversary class as in our first adversary class example from Section 2.4.1.

4.3 Efficient Guarantees for (Node) Budget Adversaries

We have shown that the anonymity impact(s) of Section 4.1 correctly characterize the advantage of any adversary. So far, we can only efficiently calculate the impact of an adversary, if all of its choices (on the senders, recipients and the compromisation strategy) are known. Moreover, we may only be able to efficiently calculate the impact, if the number of possible choices is limited.

However, as our goals are to give a wide range of anonymity guarantees for Tor, we require more efficient approaches. In this section we provide a heuristic algorithm for computing guarantees. We will show this algorithm to be sound, however, we will trade in precision for the efficiency we seek.

To this end, we define different classes of structural adversaries that statically compromise a certain subset of Tor nodes. For conveniently reasoning about different such adversaries in a unified manner, we define the concept of a *budget adversary.*

Intuitively, given a *cost function* $f\colon \mathcal{N} \to \mathbb{N} \cup \{\infty\}$ and a *budget* $B \in \mathbb{N}$, an adversary is called a *budget adversary* A_f^B if it can compromise arbitrary sets of Tor nodes $N \subseteq \mathcal{N}$, as long as $\sum\limits_{\mathsf{n} \in N} f(\mathsf{n}) \leq B$.

Formally, we define a budget adversary as a PnP adversary as follows.

Definition 4.3.1 (AnoA Budget Adversary)**.** *Consider a cost function $f\colon \mathcal{N} \to \mathbb{N} \cup \{\infty\}$ and a budget $B \in \mathbb{N}$.*

Then for any arbitrary machine M, a budget adversary A_f^B is defined as the machine $\mathrm{P{\small N}P}_{M,F_{\mathtt{block}},S_P}$*, where the predicate P on input (N, L) is defined as follows:*

- *If $L \neq \emptyset$, $P(N, L) = \mathtt{false}$.*
- *If $\sum_{\mathsf{n} \in N} f(\mathsf{n}) > B$, $P(N, L) = \mathtt{false}$.*
- *Otherwise, $P(N, L) = \mathtt{true}$.*

We let $\mathcal{A}_f^B$ denote the class of all budget adversaries for f and B.

We provide several instantiations for budget adversaries.

Definition 4.3.2 (k-collusion Adversary)**.** *A* k-collusion adversary *is a budget adversary $\mathrm{A}_{f_{\mathtt{coll}}}^k$ that compromises up to k nodes of its choice, i.e., $f_{\mathtt{coll}}(\mathsf{n}) := 1$ for $n \in \mathcal{N}$.*

Definition 4.3.3 (Predicate Adversary)**.** *A* predicate adversary *is a budget adversary $\mathrm{A}_{f_P}^1$ that compromises all nodes that fulfill a given predicate P, i.e., $f_P(\mathsf{n}) := 0$ if $P(\mathsf{n}) = \mathtt{true}$, and $f_P(\mathsf{n}) := \infty$ otherwise.*

Examples of predicate adversaries include *geographic adversaries* that compromise all nodes within a certain country or a collaboration of countries, *Tor-Version adversaries* that can exploit vulnerabilities of specific versions of the Tor software and can compromise all nodes that run this version, and *subnet adversaries* that compromise all Tor nodes within a specific IP-subnet.

Definition 4.3.4 (Bandwidth Adversary). *A resource-constrained bandwidth adversary, or* bandwidth adversary *for short, is a budget adversary* $\mathrm{A}^B_{f_{\mathtt{BW}}}$ *that compromises an arbitrary set of Tor nodes with at most an overall bandwidth of* B*, i.e., for* $n \in \mathcal{N}$*, we have* $f_{\mathtt{BW}}(\mathsf{n}) := \mathsf{n}.\mathtt{BW}$ *for* $n \in \mathcal{N}$*, where* $\mathsf{n}.\mathtt{BW}$ *denotes the bandwidth of node* n*.*

This adversary model allows us to provide anonymity bounds in the presence of adversaries that manage to observe a certain percentage of all traffic within the Tor network, e.g., by adding fake nodes or by assuming control over existing nodes.

Definition 4.3.5 (Monetary Adversary). *A* monetary adversary *is a budget adversary* $\mathrm{A}^B_{f_\$}$ *that compromises Tor nodes with a monthly monetary maintenance and renting cost of at most* B*. For a set of providers* $\mathcal{P}$ *and a function* $\mathtt{price}\colon \mathcal{P} \times \mathbb{N} \to \mathbb{N}$ *that assigns a price for each provider and offered bandwidth, we have* $f_\$(\mathsf{n}) := \mathtt{price}(\mathsf{n}.\mathtt{provider}, \mathsf{n}.\mathtt{BW})$ *for* $n \in \mathcal{N}$*, where* $\mathsf{n}.\mathtt{provider}$ *and* $\mathsf{n}.\mathtt{BW}$ *denotes the provider and the bandwidth of node* n*, respectively.*

Monetary adversaries reflect adversaries with a limited budget for the operational cost of running Tor nodes.

4.3.1 Anonymity Impact of a Budget Adversary

We now combine our formalization of observations and of their anonymity impact from Chapter 3 with our definition of a budget adversary. Thereby, we compute a bound on the anonymity impact of a budget adversary on Tor's path selection algorithm for each of the three considered anonymity notions.

A straight-forward method for computing a bound on the impact of a budget adversary is to enumerate all possible sets of nodes that this adversary could compromise, to compute the observational impact of every one of these sets, and to output the maximal such impact. This method, however, requires a huge computational effort, as the number of possible sets is exponential in the number of Tor nodes for most budget adversaries. Consequently, we derive a methodology to soundly calculate the anonymity impact of each individual Tor node and then use these to derive an overall bound.

Definition 4.3.6 (Modified Observation Impact). *Let* $\mathsf{S}_0, \mathsf{S}_1 \in \mathcal{S}$ *denote two senders, let* $\mathsf{R}_0, \mathsf{R}_1 \in \mathcal{R}$ *denote two recipients, let* $N \subseteq \mathcal{E}$ *denote a set of compromised Tor entities, let* PS *denote a path selection algorithm, and let* α_X *for* $X \in \{\mathsf{SA}, \mathsf{RA}, \mathsf{REL}\}$ *be an anonymity notion. Then, as we consider* $\varepsilon = 0$*,* $\mathtt{impact}^{\mathcal{O}}_X(N)$*, as defined in Figure 4.8, denotes the* observation impact *of* N *for* α_X *and* PS$, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1$*, where* $\mathsf{Obs}_{\not\perp} = \mathcal{O} \setminus \{(\perp,\perp,\perp,\perp,\perp)\}$ *is the set of all observations in which at least one position is different from* $\perp$ *and where* $\mathsf{Obs}_2 = \{\mathsf{S}_0,\mathsf{S}_1\} \times \mathcal{N}^3 \times \{\perp\} \cup \{\perp\} \times \mathcal{N}^3 \times \{\mathsf{R}_0,\mathsf{R}_1\} \cup \{\mathsf{S}_0,\mathsf{S}_1\} \times \mathcal{N}^3 \times \{\mathsf{R}_0,\mathsf{R}_1\}$ *is the set of observations that are made by two or more compromised nodes in a circuit.*

$$\begin{aligned}
\texttt{impact}^{\mathcal{O}}_{\mathsf{SA}}(N) := \sum_{o \in \mathsf{Obs}_{\not\perp}} \phi\Big(\quad & \texttt{Pr}\left[o = \textsc{obs}[N](c), c \leftarrow \textsc{ps}(\mathsf{S}_0, \mathsf{R}_0)\right], \\
& \texttt{Pr}\left[o = \textsc{obs}[N](c), c \leftarrow \textsc{ps}(\mathsf{S}_1, \mathsf{R}_0)\right] \Big); \\
\texttt{impact}^{\mathcal{O}}_{\mathsf{RA}}(N) := \sum_{o \in \mathsf{Obs}_{\not\perp}} \phi\Big(\quad & \texttt{Pr}\left[o = \textsc{obs}[N](c), c \leftarrow \textsc{ps}(\mathsf{S}_0, \mathsf{R}_0)\right], \\
& \texttt{Pr}\left[o = \textsc{obs}[N](c), c \leftarrow \textsc{ps}(\mathsf{S}_0, \mathsf{R}_1)\right] \Big); \\
\texttt{impact}^{\mathcal{O}}_{\mathsf{REL}}(N) := \sum_{o \in \mathsf{Obs}_{\not\perp}} \phi\Big(\quad & \big(\texttt{Pr}\left[o = \textsc{obs}[N](c), c \leftarrow \textsc{ps}(\mathsf{S}_0, \mathsf{R}_0)\right] \\
& + \texttt{Pr}\left[o = \textsc{obs}[N](c), c \leftarrow \textsc{ps}(\mathsf{S}_1, \mathsf{R}_1)\right] \big)/2, \\
& \big(\texttt{Pr}\left[o = \textsc{obs}[N](c), c \leftarrow \textsc{ps}(\mathsf{S}_0, \mathsf{R}_1)\right] \\
& + \texttt{Pr}\left[o = \textsc{obs}[N](c), c \leftarrow \textsc{ps}(\mathsf{S}_1, \mathsf{R}_0)\right] \big)/2 \Big).
\end{aligned}$$

Figure 4.8: Slightly modified definition of $\texttt{impact}^{\mathcal{O}}_{X}(N)$ instead of $\varepsilon\texttt{-impact}^{\mathcal{O}}_{X}(N)$, where we set $\varepsilon = 0$ and exclude empty observations.

For singletons $N = \{n\}$, we write $\texttt{impact}^{\mathcal{O}}_{X}(n)$ instead of $\texttt{impact}^{\mathcal{O}}_{X}(\{n\})$. For the sake of readability, we did not explicitly include $\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1$ and $\textsc{ps}$ as additional parameters of $\texttt{impact}^{\mathcal{O}}_{X}(N)$ but consider them clear from the context, similarly in the upcoming definitions. Where necessary to clarify $\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1$ we write $\texttt{impact}^{\mathcal{O}}_{X;\mathsf{S}_0,\mathsf{S}_1,\mathsf{R}_0,\mathsf{R}_1}(N)$ for this and upcoming definitions.

We distinguish two kinds of impact in the following that contribute to the overall anonymity impact of a Tor node: *direct anonymity impact* and *indirect anonymity impact.* Direct impact considers those information gained from observing compromised and honest Tor entities. Indirect impact is more subtle: it captures what the adversary can additionally learn from the *absence* of observations, i.e., which compromised Tor nodes were *not* used in the circuit, and from which it can hence draw corresponding conclusions. In the following, we elaborate on both cases separately first.

Direct Anonymity Impact The direct anonymity impact of a Tor node represents the observation impact (Definition 4.3.6) of this node in all circuits in which it is present, i.e., of all observations that contain the node. For sender anonymity and recipient anonymity, we consider only one node at a time, whereas for relationship anonymity, we also consider observations made by pairs of nodes. In the following, we explain this strategy for each of the anonymity notions.

For the direct anonymity impact on sender anonymity, a compromised guard node is sufficient. Consequently, each direct impact made by a compromised guard node in combination with any other node is also already made by this guard node alone. Each observation made by a compromised middle node in combination with the compromised recipient equals the observation made by a compromised middle node and a compromised exit node in the same circuit. Thus, for the direct impact it suffices to consider each compromised node n individually by calculating $\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{n})$. For the direct anonymity impact on recipient anonymity, the reasoning is analogous to the reasoning for sender anonymity, where for every node n we now calculate $\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n})$. For relationship anonymity, the observations made by all three nodes in a circuit equals the observation made by the guard node and the exit node. Thus, it suffices to consider all observations in which one or two nodes are compromised. We divide the set of all observations into the set of observations made by individual nodes and the observations made by two nodes. For the first set of observations, analogously to sender anonymity and recipient anonymity, we consider each compromised n individually by calculating $\mathtt{impact}^{\mathcal{O}}_{\mathsf{REL}}(\mathsf{n})$. For additionally considering the second set of observations, we define two helper functions $\mathtt{impact}^{\mathcal{O}\text{-}2}_{\mathsf{REL}}$ and $\mathtt{impact}^{\mathtt{combined}}_{\mathsf{REL}}$, see Figure 4.9. The intuition behind these functions is as follows:

- $\mathtt{impact}^{\mathcal{O}\text{-}2}_{\mathsf{REL}}(\{\mathsf{n},\mathsf{n}'\})$ defines the direct impact that a pair of nodes n and n' have on relationship anonymity. Formally, this is defined as the observation impact of $\{\mathsf{n},\mathsf{n}'\}$ over all observations $o \in \mathsf{Obs}_2$, i.e., all observations that are made by two or more compromised nodes in a circuit.

- $\mathtt{impact}^{\mathtt{combined}}_{\mathsf{REL}}(\mathsf{n})$ soundly combines the observational impact $\mathtt{impact}^{\mathcal{O}}_{\mathsf{REL}}(\mathsf{n})$ of n individually with a worst-case approximation over the direct impact of every pair of n with other compromised nodes n'.

Indirect Anonymity Impact The indirect anonymity impact of a Tor node represents the observation impact of this node on all circuits in which it is not present. The observation that a compromised node is not present in a circuit can significantly modify the anonymity impact of an observation, as illustrated by the following example.

Example 4.3.1 (Indirect Impact). *Consider the following set of circuits:* $X_{\mathsf{S}_0,\mathsf{n_g},\mathsf{n_m}} := \{c = (\mathsf{S},\mathsf{n_g}',\mathsf{n_m}',\mathsf{n_x}',\mathsf{R}) \mid \mathsf{n_g}' = \mathsf{n_g} \wedge \mathsf{n_m}' = \mathsf{n_m} \wedge \mathsf{S} = \mathsf{S}_0\}$ *for a guard node* $\mathsf{n_g}$ *and a middle node* $\mathsf{n_m}$*. On each circuit* $\mathsf{C} \in X_{\mathsf{S},\mathsf{n_g},\mathsf{n_m}}$*, an adversary that only compromises the guard makes the observation* $o := \textsc{obs}[\{\mathsf{n_g}\}](c) = (\mathsf{S}_0,\mathsf{n_g},\mathsf{n_m},\bot,\bot)$ *independently of the exit node and the recipient. By adding an exit node* $\mathsf{n_x}$ *to the set of compromised nodes, the probability for the observation o changes. Whenever* $\mathsf{n_x}$ *is the exit node of the circuit, instead of o the adversary will make the observation* $(\mathsf{S}_0,\mathsf{n_g},\mathsf{n_m},\mathsf{n_x},\mathsf{R})$*. If the compromised exit node is more likely to be used for one specific recipient* R*, then not observing the exit node in this scenario leaks information about the recipient.*

We consequently capture the impact of unobserved compromised nodes by adding the impact of their absence to any given observation made by another compromised node. To formally define the indirect impact $\mathtt{impact}_X^{\mathtt{ind}}$ for anonymity notion α_X, we define several helper functions for computing $\mathtt{impact}_X^{\mathtt{ind}}$, see Figure 4.10. The intuition behind these functions is as follows:

- $\mathtt{impact}_{\mathtt{indirect}}^{(ab,cd)}(\mathsf{n_g}, \mathsf{n_x})$ defines the mutual indirect impact that a compromised guard node $\mathsf{n_g}$ has on the observations of a compromised exit node $\mathsf{n_x}$, and vice versa.
- $\mathtt{impact}_{\mathtt{Rec1}}$ and $\mathtt{impact}_{\mathtt{Rec2}}$ describe the impact of compromised nodes on observations made by a compromised recipient (as used for sender anonymity): The first notion $\mathtt{impact}_{\mathtt{Rec1}}$ considers the impact that any individual (compromised) node may have on the (lack of) observations of a compromised recipient; the second one $\mathtt{impact}_{\mathtt{Rec2}}$ bounds the maximal error in calculating the first one.
- $\mathtt{impact}_{\mathtt{Sen1}}$ and $\mathtt{impact}_{\mathtt{Sen2}}$ analogously describe the impact of compromised nodes on observations made by a compromised sender (as used for recipient anonymity).

Based on these helper functions, the indirect impact $\mathtt{impact}_X^{\mathtt{ind}}$ is defined in Figure 4.11. We explain this definition for all three cases of X:

Indirect anonymity impact for sender anonymity. $\mathtt{impact}_{\mathsf{SA}}^{\mathtt{ind}}$ is computed by combining the indirect impact of compromised guard nodes on observations made by exit nodes ($\mathtt{impact}_{\mathtt{indirect}}^{(10,00)}$) with the impact of compromised guard nodes and middle nodes on the observations made by the recipient alone ($\mathtt{impact}_{\mathtt{Rec1}}$ and $\mathtt{impact}_{\mathtt{Rec2}}$). Whenever a compromised node is more likely to be used as a guard node (or middle node) by the sender S_1, then *not* observing this node is is more likely when the sender is S_0.

Indirect anonymity impact for recipient anonymity. $\mathtt{impact}_{\mathsf{RA}}^{\mathtt{ind}}$ is computed by combining the indirect impact of compromised exit nodes on observations made by guard nodes ($\mathtt{impact}_{\mathtt{indirect}}^{(01,00)}$) with the impact of compromised middle nodes and exit nodes on the observations made by observing the sender alone ($\mathtt{impact}_{\mathtt{Sen1}}$ and $\mathtt{impact}_{\mathtt{Sen2}}$). Whenever a compromised node is more likely to be used as an exit node (or middle node) for contacting the recipient R_1, then *not* observing this node is is more likely when the recipient is R_0.

Indirect anonymity impact for relationship anonymity. $\mathtt{impact}_{\mathsf{REL}}^{\mathtt{ind}}$ is computed by combining all possible ways in which compromised guard nodes can impact observations of compromised exit nodes and vice versa. The observations of compromised guard nodes contain the sender of a communication and thus the adversary only needs to find out the recipient, making these cases comparable to the case of recipient anonymity; the observations of compromised exit nodes contain the recipient of a communication and thus the adversary only needs to find out the sender of a communication, making these cases comparable to sender anonymity. Since for relationship anonymity the

$$\mathtt{impact}^{\mathtt{combined}}_{\mathsf{REL}}(\mathsf{n}, \mathrm{A}^B_f) := \mathtt{impact}^{\mathcal{O}}_{\mathsf{REL}}(\mathsf{n}) + \max_{K \overset{B-f(\mathsf{n}),f}{\subseteq} \mathcal{N}} \Big(\sum_{\mathsf{m}\in K} \mathtt{impact}^{\mathcal{O}\text{-}2}_{\mathsf{REL}}(\{\mathsf{n},\mathsf{m}\}) \Big)$$

$$\begin{aligned}\mathtt{impact}^{\mathcal{O}\text{-}2}_{\mathsf{REL}}(N) := \sum_{o\in\mathsf{Obs}_2} \phi\Big(\quad & \big(\mathtt{Pr}\left[o = \textsc{obs}[N](c); c \leftarrow \textsc{ps}(\mathsf{S}_0,\mathsf{R}_0)\right] \\ & + \mathtt{Pr}\left[o = \textsc{obs}[N](c); c \leftarrow \textsc{ps}(\mathsf{S}_1,\mathsf{R}_1)\right]\big)/2, \\ & \big(\mathtt{Pr}\left[o = \textsc{obs}[N](c); c \leftarrow \textsc{ps}(\mathsf{S}_0,\mathsf{R}_1)\right] \\ & + \mathtt{Pr}\left[o = \textsc{obs}[N](c); c \leftarrow \textsc{ps}(\mathsf{S}_1,\mathsf{R}_0)\right]\big)/2\Big)\end{aligned}$$

where the set of observations that are made by two or more compromised nodes in a circuit is defined as $\mathsf{Obs}_2 = \{\mathsf{S}_0,\mathsf{S}_1\} \times \mathcal{N}^3 \times \{\bot\} \cup \{\bot\} \times \mathcal{N}^3 \times \{\mathsf{R}_0,\mathsf{R}_1\} \cup \{\mathsf{S}_0,\mathsf{S}_1\} \times \mathcal{N}^3 \times \{\mathsf{R}_0,\mathsf{R}_1\}$

Figure 4.9: Helper functions for the direct impact of nodes, as used in Figure 4.12.

set of compromised entities N neither contains any of the senders or recipients of the challenge, the indirect impacts of senders and recipients do not apply here.

Defining the Anonymity Impact We finally give the overall definition of *anonymity impact*; we will explain it for the individual anonymity notions in detail after the definition.

Definition 4.3.7 (Anonymity Impact)**.** *Let* $\mathsf{S}_0, \mathsf{S}_1$ *be two senders, let* $\mathsf{R}_0, \mathsf{R}_1$ *be two recipients, let* $\textsc{ps}$ *be a path selection algorithm, let* α_X *for* $X \in \{\mathsf{SA}, \mathsf{RA}, \mathsf{REL}\}$ *be an anonymity notion, and let* A^B_f *be a budget adversary. Then,* $\mathtt{impact}_X(\mathrm{A}^B_f)$*, as defined in Figure 4.12, defines the* anonymity impact *of* A^B_f *for* α_X *and* $\textsc{ps}, \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1$.

The computation of the anonymity impact $\mathtt{impact}_X(\mathrm{A}^B_f)$ depends on the considered anonymity notion α_X as follows.

Sender anonymity. The impact of any budget adversary A^B_f on sender anonymity constitutes at most the aggregated observation impact of the optimal set of compromised nodes together with the observation impact $\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{R}_0)$ of the compromised recipient R_0 (which is assumed for sender anonymity) and the indirect impact on sender anonymity, c.f., Equations (1) in Figures 4.11 and 4.12.

Recipient anonymity. Analogously, the impact of any budget adversary A^B_f on recipient anonymity is, at most, the aggregated observation impact of the optimal set of

$$
\begin{aligned}
\mathtt{impact}^{(ab)(cd)}_{\mathtt{indirect}}(\mathsf{n_g}, \mathsf{n_x}) &:= \sum_{\mathsf{n_m}\in\mathcal{N}} \phi(P^{ab}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, P^{cd}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}})\\
\mathtt{impact}_{\mathtt{Rec1}}(\mathsf{n}) &:= \sum_{\mathsf{n_x}\in\mathcal{N}} \phi(\sum_{\mathsf{n'}\in\mathcal{N}} (P^{10}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}} + P^{10}_{\mathsf{n'},\mathsf{n},\mathsf{n_x}}), \sum_{\mathsf{n'}\in\mathcal{N}} (P^{00}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}} + P^{00}_{\mathsf{n'},\mathsf{n},\mathsf{n_x}}))\\
\mathtt{impact}_{\mathtt{Rec2}}(\mathsf{n}, \mathsf{n'}) &:= \sum_{\mathsf{n_x}\in\mathcal{N}} \phi(P^{00}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}}, P^{10}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}})\\
\mathtt{impact}_{\mathtt{Sen1}}(\mathsf{n}) &:= \sum_{\mathsf{n_g}\in\mathcal{N}} \phi(\sum_{\mathsf{n'}\in\mathcal{N}} (P^{01}_{\mathsf{n_g},\mathsf{n},\mathsf{n'}} + P^{01}_{\mathsf{n_g},\mathsf{n'},\mathsf{n}}), \sum_{\mathsf{n'}\in\mathcal{N}} (P^{00}_{\mathsf{n_g},\mathsf{n},\mathsf{n'}} + P^{00}_{\mathsf{n_g},\mathsf{n'},\mathsf{n}}))\\
\mathtt{impact}_{\mathtt{Sen2}}(\mathsf{n}, \mathsf{n'}) &:= \sum_{\mathsf{n_g}\in\mathcal{N}} \phi(P^{00}_{\mathsf{n_g},\mathsf{n},\mathsf{n'}}, P^{01}_{\mathsf{n_g},\mathsf{n},\mathsf{n'}})
\end{aligned}
$$

$$
\begin{aligned}
\text{where} \qquad & N \overset{B,f}{\subseteq} \mathcal{N} := N \subseteq \mathcal{N} s.t. \sum_{\mathsf{n}\in N} f(\mathsf{n}) \leq B\\
& P^{ab}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} := \mathtt{Pr}\left[(\mathsf{S}_a, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_b) = c; c \leftarrow \textsc{ps}(\mathsf{S}_a, \mathsf{R}_b)\right]
\end{aligned}
$$

Figure 4.10: Helper functions for the indirect impact of nodes, as used in Figure 4.11.

compromised nodes together with the observation impact $\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{S}_0)$ of the compromised sender S_0 (which is assumed for recipient anonymity) and the indirect impact on recipient anonymity, c.f., Equations (2) in Figures 4.11 and 4.12.

Relationship anonymity. Analogously, the impact on any budget adversary A^B_f on relationship anonymity is, at most, the aggregated observation impact of the optimal set of compromised nodes and the indirect impact on relationship anonymity, c.f., Equations (3) in Figures 4.11 and 4.12. In contrast to sender anonymity and recipient anonymity, we need to consider a combination of nodes for the direct impact of relationship anonymity, see, $\mathtt{combined}_{\mathsf{REL}}$ in Figure 4.10.

For relationship anonymity we technically assume that the empty observation $o = (\bot, \bot, \bot, \bot, \bot)$ has an anonymity impact of zero, while practically capturing the empty observation by calculating the anonymity impact in both directions, i.e., not only computing a guarantee for the scenarios with $b = 0$ (S_0 communicates with R_0 or S_1 communicates with R_1) against the scenarios with $b = 1$ (S_0 communicates with R_1 or S_1 communicates with R_0), but also *vice versa*.

Precision of our Calculation For all circuit observations made by sets of compromised nodes, individual nodes, pairs of nodes, and sender or recipient, our calculation

$$
\begin{aligned}
&(1)\ \texttt{impact}^{\texttt{ind}}_{\mathsf{SA}}(\mathsf{n}, \mathrm{A}^B_f) := \texttt{impact}_{\texttt{Rec1}}(\mathsf{n}) \\
&\quad + \max_{N \overset{B-f(\mathsf{n}),f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}' \in N} \left(\texttt{impact}^{(10)(00)}_{\texttt{indirect}}\left(\mathsf{n}, \mathsf{n}'\right) + \texttt{impact}_{\texttt{Rec2}}(\mathsf{n}, \mathsf{n}')\right) \\
&(2)\ \texttt{impact}^{\texttt{ind}}_{\mathsf{RA}}(\mathsf{n}, \mathrm{A}^B_f) := \texttt{impact}_{\texttt{Sen1}}(\mathsf{n}) \\
&\quad + \max_{N \overset{B-f(\mathsf{n}),f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}' \in N} \left(\texttt{impact}^{(01)(00)}_{\texttt{indirect}}\left(\mathsf{n}', \mathsf{n}\right) + \texttt{impact}_{\texttt{Sen2}}(\mathsf{n}, \mathsf{n}')\right) \\
&(3)\ \texttt{impact}^{\texttt{ind}}_{\mathsf{REL},\mathsf{S}_a,\mathsf{S}_c,\mathsf{R}_b,\mathsf{R}_d}(\mathsf{n}, \mathrm{A}^B_f) := \max_{K \overset{B-f(\mathsf{n}),f}{\subseteq} \mathcal{N} \setminus \{\mathsf{n}\}} \sum_{\mathsf{n}' \in N} \left(\frac{1}{2}\left(\texttt{impact}^{(ad)(ab)}_{\texttt{indirect}}\left(\mathsf{n}, \mathsf{n}'\right)\right.\right. \\
&\qquad + \texttt{impact}^{(cb)(cd)}_{\texttt{indirect}}\left(\mathsf{n}, \mathsf{n}'\right) \\
&\qquad + \texttt{impact}^{(ad)(cd)}_{\texttt{indirect}}\left(\mathsf{n}', \mathsf{n}\right) \\
&\qquad \left.\left. + \texttt{impact}^{(cb)(ab)}_{\texttt{indirect}}\left(\mathsf{n}', \mathsf{n}\right)\right)\right)
\end{aligned}
$$

Figure 4.11: Definition of indirect impact (for Definition 4.3.7)

$$
\begin{aligned}
&(1)\ \texttt{impact}_{\mathsf{SA}}(\mathrm{A}^B_f) := \texttt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{R}_0) \\
&\quad + \max_{N \overset{B,f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n} \in N} \left(\texttt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{n}) + \texttt{impact}^{\texttt{ind}}_{\mathsf{SA}}(\mathsf{n}, \mathrm{A}^B_f)\right) \\
&(2)\ \texttt{impact}_{\mathsf{RA}}(\mathrm{A}^B_f) := \texttt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{S}_0) \\
&\quad + \max_{N \overset{B,f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n} \in N} \left(\texttt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n}) + \texttt{impact}^{\texttt{ind}}_{\mathsf{RA}}(\mathsf{n}, \mathrm{A}^B_f)\right) \\
&(3)\ \texttt{impact}_{\mathsf{REL}}(\mathrm{A}^B_f) := \max\left(\delta_{\mathsf{REL}}(\mathrm{A}^B_f, 0, 1), \delta_{\mathsf{REL}}(\mathrm{A}^B_f, 1, 0)\right) \\
&\text{where } \delta_{\mathsf{REL}}(\mathrm{A}^B_f, b, d) := \max_{N \overset{B,f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n} \in N} \left(\texttt{impact}^{\texttt{combined}}_{\mathsf{REL},\mathsf{S}_0,\mathsf{S}_1,\mathsf{R}_b,\mathsf{R}_d}(\mathsf{n}, \mathrm{A}^B_f)\right. \\
&\qquad \left. + \texttt{impact}^{\texttt{ind}}_{\mathsf{REL},\mathsf{S}_0,\mathsf{S}_1,\mathsf{R}_b,\mathsf{R}_d}(\mathsf{n}, \mathrm{A}^B_f)\right)
\end{aligned}
$$

Figure 4.12: Definition of $\texttt{impact}_X(\mathrm{A}^B_f)$, in order to define anonymity impact (Definition 4.3.7)

of $\texttt{impact}^{\mathcal{O}}_{\mathsf{SA}}(N)$, $\texttt{impact}^{\mathcal{O}}_{\mathsf{RA}}(N)$ and $\texttt{impact}^{\mathcal{O}}_{\mathsf{REL}}(N)$ precisely captures the anonymity impact for the respective notion. However, when we aggregate the impact of individual nodes in $\texttt{impact}_X$ in order to derive our overall bounds for the anonymity impact of a budget adversary, we might count observations made for the same circuit more than once, and therefore over-approximate the impact of the individual observations. Moreover, our bound on the (indirect) impact of nodes soundly overestimates the impact. We decided to accept this slight over-approximation for reasons of performance and scalability, as it allows us to compute bounds for budget adversaries based on each node individually; otherwise, we would have to combine all possible observations of all subsets of the set of nodes that fall within the budget. We refer to Section 5.1.5 for an estimation of the precision. Furthermore, we assume that an adversary can mount traffic correlation attacks with perfect accuracy, i.e., whenever it observes traffic at two different points in the Tor network, we assume that the adversary can determine if this traffic belongs to the same Tor circuit. This assumption is motivated by the high accuracy achieved by recent work on traffic correlation attacks [104, 69, 85]; yet, it still constitutes an over-approximation.

4.3.2 Correctness of $\texttt{impact}_X$ bounds

We now show that $\texttt{impact}_X(\mathrm{A}_f^B)$, as defined in Definition 4.3.7, closely corresponds to the notion of adversary's advantage in the AnoA framework for budget adversaries A_f^B, thereby establishing the output of $\texttt{impact}_X$ as accurate bounds for Tor against such adversaries.

Theorem 4.3.1 (Soundness). *For every anonymity notion α_X with $X \in \{\mathsf{SA}, \mathsf{RA}, \mathsf{REL}\}$, all senders $\mathsf{S}_0, \mathsf{S}_1 \in \mathcal{S}$, all recipients $\mathsf{R}_0, \mathsf{R}_1 \in \mathcal{R}$, every path selection algorithm* PS*, and every budget B and every cost function f, Tor is $(\alpha_X, 1, \varepsilon = 0, \delta)$-IND-CDP for the class of budget adversaries A_f^B, where $\delta = \texttt{impact}_X(\mathcal{A}_f^B)$, as calculated in Section 4.3, up to a negligible additive factor.*

Since the proof of this theorem is involved, we dedicate the following subsections to it. We start with an overall proof outline. After that, we define the *visible elements* and the *core* of an observation, i.e., the nodes that are visible to the adversary and nodes that lead to the observation, and we describe the probability to make an observation in terms of the core and all non-core nodes (Section 4.4.2). Using these general concepts, we show the theorem separately for the three anonymity notions (Sections 4.4.3 to 4.4.5). Finally, we show that approximating the impacts via local maximization of the impacts is a sound over-approximation (Section 4.4.6).

The empty observation Although we exclude the empty observation $(\bot, \bot, \bot, \bot, \bot)$ from our computations, we do not exclude it from the guarantee itself. An adversary with some background knowledge that knows about the existence of a Tor circuit, but that makes no observation at all can still learn that none of its compromised Tor nodes were used in the circuit, which might be more likely for one of the scenarios. For sender anonymity (and for recipient anonymity) we do not need to consider this case, as we can soundly assume that the recipient (or the sender) is always compromised, which always leads to a non-empty observation. For relationship we technically assume that the empty observation has an anonymity impact of zero, while practically capturing the empty observation by calculating the anonymity impact in both directions, i.e., not only computing a guarantee for the scenarios with $b = 0$ (S_0 communicates with R_0 or S_1 communicates with R_1) against the scenarios with $b = 1$ (S_0 communicates with R_1 or S_1 communicates with R_0), but also *vice versa*.

4.4 Proof of Soundness

We start with an overall proof outline. After that, we define the *visible elements* and the *core* of an observation, i.e., the nodes that are visible to the adversary and nodes that lead to the observation, and we describe the probability to make an observation in terms of the core and all non-core nodes (Section 4.4.2). Using these general concepts, we show the theorem separately for the three anonymity notions (Sections 4.4.3 to 4.4.5). Finally, we show that approximating the impacts via local maximization of the impacts is a sound over-approximation (Section 4.4.6).

4.4.1 Overall Proof Outline

By Theorem 3.3.1 we know that it suffices to analyze the specialized AnoA game from Section 3.3. By Theorem 4.1.1 we know that any observation made by any set of Tor nodes (in combination with senders and recipients) N impacts anonymity by exactly $\texttt{impact}_X^{\mathcal{O}}(N)$ for the anonymity notion α_X under consideration. Intuitively, the remaining proof of soundness divides the set of all observations into distinct subsets of observations, depending on where compromised nodes are located in a circuit. Then, for every such set, we compare the impact of each observation if a set of Tor entities is compromised with the sum of the impacts of all compromised Tor entities on their own. Since we sum over all these Tor entities, for the majority of observations the impact of the sum is larger than the impact of the combined set. However, the lack of observation of certain (compromised) Tor nodes can increase the impact of observations made by other compromised Tor entities. In the remainder of this section, we will formally substantiate these claims.

A Simple Lemma For our proofs concerning the soundness of our bounds we will need the following simple lemma. Note that we write ϕ for ϕ_ε, as $\varepsilon = 0$ will hold for the remainder of this chapter.

Lemma 4.4.1 (Properties of ϕ). *The function ϕ has the following properties:*

(i) For all $a, b, c, d \geq 0$, we have $\phi(a + b, c + d) \leq \phi(a, c) + \phi(b, d)$.

(ii) For all $a, b, c, d \geq 0$, we have $\phi(a - b, c - d) \leq \phi(a, c) + \phi(d, b)$.

Proof. We show each property via a case distinction over the two cases of the conditional within ϕ.

(i) Let a, b, c, d. We distinguish two cases:

- **Case 1:** $(a+b) \leq (c+d)$. Then $\phi(a+b, c+d) = 0 \leq 0+0 \leq \phi(a,c)+\phi(b,d)$.
- **Case 2:** $(a+b) > (c+d)$. Then

$$\begin{aligned}&\phi(a+b, c+d) = a+b-(c+d)\\ =&a-c+b-d \leq \phi(a,c)+\phi(b,d),\end{aligned}$$

since by definition $\phi(X,Y) \geq X-Y$.

(ii) Let $a, b, c, d \geq 0$. We distinguish two cases:

- **Case 1:** $(a-b) \leq (c-d)$. Then $\phi(a+b, c+d) = 0 \leq 0+0 \leq \phi(a,c)+\phi(d,b)$.
- **Case 2:** $(a-b) > c-d$. Then

$$\begin{aligned}&\phi(a-b, c-d) = a-b-(c-d) = a-c+d-b\\ \leq&\phi(a,c)+\phi(b,d),\end{aligned}$$

since by definition $\phi(X,Y) \geq X-Y$.

□

Additional Notation for the Proof In what follows, we write $P^{ab}[C]$ as a shortcut for $\texttt{Pr}[\mathsf{C} \in C; \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_a, \mathsf{R}_b)]$. Moreover, for the probabilities we encounter when analyzing relationship anonymity we write $P^{ab,cd}[C]$ instead of $\frac{1}{2}\left(P^{ab}[C] + P^{cd}[C]\right)$. For an observation $o = (n_1, n_2, n_3, n_4, n_5)$ we refer to the element n_i by writing $o.n_i$. For any three Tor nodes $\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}$ we write $P^{ab}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}$ for

$$\texttt{Pr}[\mathsf{C} = (\mathsf{S}_a, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_b); \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_a, \mathsf{R}_b)].$$

.

4.4.2 Visible Nodes, Observation Core and Blank Observations

We continue by defining *natural* observations that occur if the adversary can only compromise Tor entities (and not communication links between them) and by then classifying them. For every (natural) observation we define which elements are visible to the adversary. We will see that for every natural observation there is one or more sets of *core* entities that have to be compromised in order to make the respective observation. Along with these core entities we define for every observation which roles other Tor nodes have to play in order to not invalidate the observation. To this end, we define *blanked* observations, i.e., observations that only describe the type of the natural observation.

Definition 4.4.1 (Natural Observations; Visible Elements; Core Elements; Blanked Observations)**.** *We define the set* NatObs *of natural observations as* $(\mathcal{S} \times \mathcal{N} \times \{\bot\}^3) \cup (\mathcal{S} \times \mathcal{N}^2 \times \{\bot\}^2) \cup (\{\bot\} \times \mathcal{N}^3 \times \{\bot\}) \cup (\{\bot\}^2 \times \mathcal{N}^2 \times \mathcal{R}) \cup (\{\bot\}^3 \times \mathcal{N} \times \mathcal{R}) \cup (\mathcal{S} \times \mathcal{N}^3 \times \{\bot\}) \cup (\{\bot\} \times \mathcal{N}^3 \times \mathcal{R}) \cup (\mathcal{S} \times \mathcal{N}^3 \times \mathcal{R})$.

For every observation $o = (n_1, n_2, n_3, n_4, n_5) \in \mathcal{O}$*, we define the set of* visible Tor entities $V(o)$ *as* $V(o) := \{n_i \mid n_i \neq \bot, i \in \{1,2,3,4,5\}\}$.

For every natural observation $o = (n_1, n_2, n_3, n_4, n_5) \in \mathsf{NatObs}$*, we define the* core elements $\mathtt{core}(o)$ of o *as follows:*

- *If* $o \in \mathcal{S} \times \mathcal{N} \times \{\bot\}^3$*, then* $\mathtt{core}(o) := \{\{n_1\}\}$.
- *If* $o \in \mathcal{S} \times \mathcal{N}^2 \times \{\bot\}^2$ *, then* $\mathtt{core}(o) := \{\{n_2\}\}$.
- *If* $o \in \{\bot\} \times \mathcal{N}^3 \times \{\bot\}$*, then* $\mathtt{core}(o) := \{\{n_3\}\}$.
- *If* $o \in \{\bot\}^2 \times \mathcal{N}^2 \times \mathcal{R}$*, then* $\mathtt{core}(o) := \{\{n_4\}\}$.
- *If* $o \in \{\bot\}^3 \times \mathcal{N} \times \mathcal{R}$*, then* $\mathtt{core}(o) := \{\{n_5\}\}$.
- *If* $o \in \mathcal{S} \times \mathcal{N}^3 \times \{\bot\}$*, then* $\mathtt{core}(o) := \{\{n_1, n_3\}, \{n_2, n_3\}\}$.
- *If* $o \in \{\bot\} \times \mathcal{N}^3 \times \mathcal{R}$*, then* $\mathtt{core}(o) := \{\{n_3, n_4\}, \{n_3, n_5\}\}$.
- *If* $o \in \mathcal{S} \times \mathcal{N}^3 \times \mathcal{R}$*, then* $\mathtt{core}(o) := \{\{n_1, n_3, n_5\}, \{n_1, n_4\}, \{n_2, n_4\}, \{n_2, n_5\}\}$.

Naturally the core elements of an observation are always visible: For every natural observation $o \in \mathsf{NatObs}$*, and for every Core* $\in \mathtt{core}(o)$*, we have Core* $\subseteq V(o)$.

For a natural observation $o = (n_1, n_2, n_3, n_4, n_5) \in \mathsf{NatObs}$*, we define the corresponding* blanked observation $\mathtt{blank}(o)$ *as* $\mathtt{blank}(o) := (n'_1, n'_2, n'_3, n'_4, n'_5)$ *where* $n'_i := n_i$ *if* $n_i = \bot$ *and* $n'_i := \_$ *otherwise for a distinguished symbol* $\_$. *We extend the standard equality on observations* o *by treating* $\_$ *as a placeholder: an observation* $o = (n_1, n_2, n_3, n_4, n_5)$ *is equal to a blank observation* $o' = (n'_1, n'_2, n'_3, n'_4, n'_5)$ *if* $n_i = n'_i$ *for all* $n'_i \neq \_$.

Definition 4.4.2 (Observation Circuits)**.** *Let* $N \subseteq \mathcal{E}$ *be a set of compromised Tor entities and* $o \in \mathcal{O}$ *be an observation. Then we define the set of* observation circuits $\mathcal{OC}(N, o)$ *of* N *and* o *as*

$$\mathcal{OC}(N, o) := \{\mathsf{C} \in \mathsf{Circuits} \mid o = \mathcal{O}[N](\mathsf{C})\}.$$

Similarly, we define the set of blanked observation circuits $\mathcal{OC}_{\mathtt{blank}}(N, o)$ *of* N *and* o *as* $\mathcal{OC}_{\mathtt{blank}}(N, o) := \{c \in \mathsf{Circuits} \mid \mathtt{blank}(o) = \mathcal{O}[N](\mathsf{C})\}$.

We first show a basic lemma that states that an observation o by a set N can be made for a given circuit precisely if for every element *Core* of the core of o, we have that (i) the same observation is made for this circuit by *Core* and (ii) and the elements in $N \setminus$ *Core* make the corresponding blanked observation $\mathtt{blank}(o)$ for this circuit.

Lemma 4.4.2 (Classification of Core). *Let $o \in \mathsf{NatObs}$ be a natural observation, let $Core \in \mathtt{core}(o)$, and let $N \subseteq \mathcal{E}$ be a set of compromised Tor entities such that $Core \subseteq N$ and let $\mathsf{C} \in \mathsf{Circuits}$ be a circuit. Then we have*

$$\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[Core](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus Core](\mathsf{C}) = \mathtt{blank}(o).$$

Proof. Let $o \in \mathsf{NatObs}$ be a natural observation, let $Core \in \mathtt{core}(o)$, and let $N \subseteq \mathcal{E}$ such that $Core \subseteq N$ and let $\mathsf{C} = (\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}) \in \mathsf{Circuits}$ be a circuit. We distinguish the following cases depending on the structure of o.

- $o \in \mathcal{S} \times \mathcal{N} \times \{\perp\}^3$. Then $o = (\mathsf{S}', \mathsf{n_g}', \perp, \perp, \perp)$ for some $\mathsf{S}' \in \mathcal{S}, \mathsf{n_g}' \in \mathcal{N}$. By definition, $\mathtt{core}(o) = \{\{\mathsf{S}'\}\}$ and thus $Core = \{\mathsf{S}'\}$; since $Core \subseteq N$ we have $\mathsf{S}' \in N$. By definition of $\textsc{obs}$, we have

  $$\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}' \wedge \{\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_b\} \cap N = \emptyset.$$

  Moreover, we have

  $$\textsc{obs}[\{\mathsf{S}'\}](\mathsf{C}) = o \Leftrightarrow \mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}'.$$

  Since $\mathtt{blank}(o) = (\_, \_, \perp, \perp, \perp)$, we have

  $$\textsc{obs}[N \setminus \{\mathsf{S}'\}](\mathsf{C}) = \mathtt{blank}(o) \Leftrightarrow \{\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}\} \cap N = \emptyset.$$

  Thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[Core](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus Core](\mathsf{C}) = \mathtt{blank}(o)$.

- $o \in \mathcal{S} \times \mathcal{N}^2 \times \{\perp\}^2$. Then $o = (\mathsf{S}', \mathsf{n_g}', \mathsf{n_m}', \perp, \perp)$ for some $\mathsf{S}' \in \mathcal{S}, \mathsf{n_g}', \mathsf{n_m}' \in \mathcal{N}$. By definition, $\mathtt{core}(o) = \{\{\mathsf{n_g}'\}\}$ and thus $Core = \{\mathsf{n_g}'\}$; since $Core \subseteq N$ we have $\mathsf{n_g}' \in N$. By definition of $\textsc{obs}$, we have

  $$\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \{\mathsf{n_m}, \mathsf{n_x}, \mathsf{R}\} \cap (N \setminus \{\mathsf{n_g}'\}) = \emptyset.$$

  Moreover, we have

  $$\textsc{obs}[\{\mathsf{n_g}'\}](\mathsf{C}) = o \Leftrightarrow \mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}'.$$

  Since $\mathtt{blank}(o) = (\_, \_, \_, \perp, \perp)$, we have

  $$\textsc{obs}[N \setminus \{\mathsf{n_g}'\}](\mathsf{C}) = \mathtt{blank}(o) \Leftrightarrow \{\mathsf{n_m}, \mathsf{n_x}, \mathsf{R}\} \cap (N \setminus \{\mathsf{n_g}'\}) = \emptyset.$$

  Thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[Core](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus Core](\mathsf{C}) = \mathtt{blank}(o)$.

- $o \in \{\perp\} \times \mathcal{N}^3 \times \{\perp\}$. Then $o = (\perp, \mathsf{n_g}', \mathsf{n_m}', \mathsf{n_x}', \perp)$ for some $\mathsf{n_g}', \mathsf{n_m}', \mathsf{n_x}' \in \mathcal{N}$. By definition, $\mathtt{core}(o) = \{\{\mathsf{n_m}'\}\}$ and thus $Core = \{\mathsf{n_m}'\}$; since $Core \subseteq N$ we have $\mathsf{n_m}' \in N$. By definition of $\textsc{obs}$, we have

  $$\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \{\mathsf{S}, \mathsf{n_g}, \mathsf{n_x}, \mathsf{R}\} \cap (N \setminus \{\mathsf{n_m}'\}) = \emptyset.$$

Moreover, we have

$$\textsc{obs}[\{\mathsf{n_m}'\}](\mathsf{C}) = o \Leftrightarrow \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}'.$$

Since $\texttt{blank}(o) = (\bot, \_, \_, \_, \bot)$, we have

$$\textsc{obs}[N \setminus \{\mathsf{n_m}'\}](\mathsf{C}) = \texttt{blank}(o) \Leftrightarrow \{\mathsf{S}, \mathsf{n_g}, \mathsf{n_x}, \mathsf{R}\} \cap (N \setminus \{\mathsf{n_m}'\}) = \emptyset.$$

Thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[\mathit{Core}](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus \mathit{Core}](\mathsf{C}) = \texttt{blank}(o)$.

- $o \in \{\bot\}^2 \times \mathcal{N}^2 \times \mathcal{R}$. Then $o = (\bot, \bot, \mathsf{n_m}', \mathsf{n_x}', \mathsf{R}')$ for some $\mathsf{n_m}', \mathsf{n_x}' \in \mathcal{N}, \mathsf{R}' \in \mathcal{R}$. By definition, $\texttt{core}(o) = \{\{\mathsf{n_x}'\}\}$ and thus $\mathit{Core} = \{\mathsf{n_x}'\}$; since $\mathit{Core} \subseteq N$ we have $\mathsf{n_x}' \in N$. By definition of $\textsc{obs}$, we have

$$\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}' \wedge \{\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}\} \cap (N \setminus \{\mathsf{n_x}'\}) = \emptyset.$$

Moreover, we have

$$\textsc{obs}[\{\mathsf{n_x}'\}](\mathsf{C}) = o \Leftrightarrow \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}'$$

Since $\texttt{blank}(o) = (\bot, \bot, \_, \_, \_)$, we have

$$\textsc{obs}[N \setminus \{\mathsf{n_x}'\}](\mathsf{C}) = \texttt{blank}(o) \Leftrightarrow \{\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}\} \cap (N \setminus \{\mathsf{n_x}'\}) = \emptyset.$$

Thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[\mathit{Core}](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus \mathit{Core}](\mathsf{C}) = \texttt{blank}(o)$.

- $o \in \{\bot\}^3 \times \mathcal{N} \times \mathcal{R}$. Then $o = (\bot, \bot, \bot, \mathsf{n_x}', \mathsf{R}')$ for some $\mathsf{n_x}' \in \mathcal{N}, \mathsf{R}' \in \mathcal{R}$. By definition, $\texttt{core}(o) = \{\{\mathsf{R}'\}\}$ and thus $\mathit{Core} = \{\mathsf{R}'\}$ and since $\mathit{Core} \subseteq N$ we have $\mathsf{R}' \in N$. By definition of $\textsc{obs}$, we have

$$\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}' \wedge \{\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}\} \cap N = \emptyset.$$

Moreover, we have

$$\textsc{obs}[\{\mathsf{R}'\}](\mathsf{C}) = o \Leftrightarrow \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}'$$

Since $\texttt{blank}(o) = (\bot, \bot, \bot, \_, \_)$, we have

$$\textsc{obs}[N \setminus \{\mathsf{R}'\}](\mathsf{C}) = \texttt{blank}(o) \Leftrightarrow \{\mathsf{S}, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}\} \cap N = \emptyset$$

Thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[\mathit{Core}](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus \mathit{Core}](\mathsf{C}) = \texttt{blank}(o)$.

- $o \in \mathcal{S} \times \mathcal{N}^3 \times \{\bot\}$. Then $o = (\mathsf{S}', \mathsf{n_g}', \mathsf{n_m}', \mathsf{n_x}', \bot)$ for some $\mathsf{S}' \in \mathcal{S}, \mathsf{n_g}', \mathsf{n_m}', \mathsf{n_x}' \in \mathcal{N}$. By definition, $\texttt{core}(o) = \{\{\mathsf{n_g}', \mathsf{n_m}'\}, \{\mathsf{S}', \mathsf{n_m}'\}\}$. We distinguish the following two cases, depending on Core.

  Case $\mathit{Core} = \{\mathsf{n_g}', \mathsf{n_m}'\}$: Since $\mathit{Core} \subseteq N$ we have $\mathsf{n_g}', \mathsf{n_m}' \in N$. By definition of $\textsc{obs}$, we have

$$\textsc{obs}[N](\mathsf{C}) = o$$

$$\Leftrightarrow \quad \mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \{\mathsf{n_x}, \mathsf{R}\} \cap (N \setminus \{\mathsf{n_g}', \mathsf{n_m}'\}) = \emptyset.$$

Moreover, we have

$$\begin{aligned} &\textsc{obs}[\{\mathsf{n_g}', \mathsf{n_m}'\}](\mathsf{C}) = o \\ \Leftrightarrow \quad &\mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}'. \end{aligned}$$

Since $\mathtt{blank}(o) = (\_, \_, \_, \_, \bot)$, we have

$$\textsc{obs}[N \setminus \{\mathsf{n_g}', \mathsf{n_m}'\}](\mathsf{C}) = \mathtt{blank}(o) \Leftrightarrow \{\mathsf{n_x}, \mathsf{R}\} \cap (N \setminus \{\mathsf{n_g}', \mathsf{n_m}'\}) = \emptyset.$$

Thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[\mathit{Core}](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus \mathit{Core}](\mathsf{C}) = \mathtt{blank}(o)$.

Case $\mathit{Core} = \{\mathsf{S}', \mathsf{n_m}'\}$: Since $\mathit{Core} \subseteq N$ we have $\mathsf{S}', \mathsf{n_m}' \in N$. By definition of $\textsc{obs}$, we have

$$\begin{aligned} &\textsc{obs}[N](\mathsf{C}) = o \\ \Leftrightarrow \quad &\mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \{\mathsf{n_x}, \mathsf{R}\} \cap (N \setminus \{\mathsf{S}, \mathsf{n_m}'\}) = \emptyset. \end{aligned}$$

Moreover, we have

$$\begin{aligned} &\textsc{obs}[\{\mathsf{S}', \mathsf{n_m}'\}](\mathsf{C}) = o \\ \Leftrightarrow \quad &\mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}'. \end{aligned}$$

Since $\mathtt{blank}(o) = (\_, \_, \_, \_, \bot)$, we have

$$\textsc{obs}[N \setminus \{\mathsf{S}', \mathsf{n_m}'\}](\mathsf{C}) = \mathtt{blank}(o) \Leftrightarrow \{\mathsf{n_x}, \mathsf{R}\} \cap (N \setminus \{\mathsf{S}, \mathsf{n_m}'\}) = \emptyset.$$

Thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[\mathit{Core}](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus \mathit{Core}](\mathsf{C}) = \mathtt{blank}(o)$.

- $o \in \{\bot\} \times \mathcal{N}^3 \times \mathcal{R}$. Then $o = (\bot, \mathsf{n_g}', \mathsf{n_m}', \mathsf{n_x}', \mathsf{R}')$ for some $\mathsf{n_g}', \mathsf{n_m}', \mathsf{n_x}' \in \mathcal{N}, \mathsf{R}' \in \mathcal{R}$. By definition, $\mathtt{core}(o) = \{\{\mathsf{n_m}', \mathsf{n_x}'\}, \{\mathsf{n_m}', \mathsf{R}'\}\}$. We distinguish the following two cases, depending on *Core*.

  Case $\mathit{Core} = \{\mathsf{n_m}', \mathsf{n_x}'\}$: Since $\mathit{Core} \subseteq N$ we have $\mathsf{n_m}', \mathsf{n_x}' \in N$. By definition of $\textsc{obs}$, we have

$$\begin{aligned} &\textsc{obs}[N](\mathsf{C}) = o \\ \Leftrightarrow &\mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}' \wedge \{\mathsf{S}, \mathsf{n_g}\} \cap (N \setminus \{\mathsf{n_m}', \mathsf{n_x}'\}) = \emptyset \end{aligned}$$

  Moreover, we have

$$\textsc{obs}[\{\mathsf{n_m}', \mathsf{n_x}'\}](\mathsf{C}) = o \Leftrightarrow \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}'$$

  Since $\mathtt{blank}(o) = (\bot, \_, \_, \_, \_)$, we have

$$\textsc{obs}[N \setminus \{\mathsf{n_m}', \mathsf{n_x}'\}](\mathsf{C}) = \mathtt{blank}(o) \Leftrightarrow \{\mathsf{S}, \mathsf{n_g}\} \cap (N \setminus \{\mathsf{n_m}', \mathsf{n_x}'\}) = \emptyset$$

Thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[\mathit{Core}](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus \mathit{Core}](\mathsf{C}) = \mathtt{blank}(o)$.

Case $\mathit{Core} = \{\mathsf{n_m}', \mathsf{R}'\}$: Since $\mathit{Core} \subseteq N$ we have $\mathsf{n_m}', \mathsf{R}' \in N$. By definition of OBS, we have

$$\begin{aligned}&\textsc{obs}[N](\mathsf{C}) = o\\ \Leftrightarrow &\mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}' \wedge \{\mathsf{S}, \mathsf{n_g}\} \cap (N \setminus \{\mathsf{n_m}', \mathsf{R}\}) = \emptyset\end{aligned}$$

Moreover, we have

$$\textsc{obs}[\{\mathsf{n_m}', \mathsf{R}'\}](\mathsf{C}) = o \Leftrightarrow \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}'$$

Since $\mathtt{blank}(o) = (\bot, \_, \_, \_, \_)$, we have

$$\textsc{obs}[N \setminus \{\mathsf{n_m}', \mathsf{R}'\}](\mathsf{C}) = \mathtt{blank}(o) \Leftrightarrow \{\mathsf{S}, \mathsf{n_g}\} \cap (N \setminus \{\mathsf{n_m}', \mathsf{R}\}) = \emptyset$$

Thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[\mathit{Core}](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus \mathit{Core}](\mathsf{C}) = \mathtt{blank}(o)$.

- $o \in \mathcal{S} \times \mathcal{N}^3 \times \mathcal{R}$. Then $o = (\mathsf{S}', \mathsf{n_g}', \mathsf{n_m}', \mathsf{n_x}', \mathsf{R}')$ for some $\mathsf{n_g}', \mathsf{n_m}', \mathsf{n_x}' \in \mathcal{N}, \mathsf{S}' \in \mathcal{S}, \mathsf{R}' \in \mathcal{R}$. By definition, $\mathtt{core}(o) = \{\{\mathsf{S}', \mathsf{n_m}', \mathsf{R}'\}, \{\mathsf{S}', \mathsf{n_x}'\}, \{\mathsf{n_g}', \mathsf{n_x}'\}, \{\mathsf{n_g}', \mathsf{R}'\}\}$. Be definition of OBS we have

$$\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}'$$

Equally, for every element $\mathit{Core} \in \mathtt{core}(o)$ we have

$$\textsc{obs}[\mathit{Core}](\mathsf{C}) = o \Leftrightarrow \mathsf{S} = \mathsf{S}' \wedge \mathsf{n_g} = \mathsf{n_g}' \wedge \mathsf{n_m} = \mathsf{n_m}' \wedge \mathsf{n_x} = \mathsf{n_x}' \wedge \mathsf{R} = \mathsf{R}'$$

Moreover, since $\mathtt{blank}(o) = (\_, \_, \_, \_, \_)$, for every observation $o' \in \mathcal{O}$ we have $o' = \mathtt{blank}(o)$ and thus, $\textsc{obs}[N](\mathsf{C}) = o \Leftrightarrow \textsc{obs}[\mathit{Core}](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus \mathit{Core}](\mathsf{C}) = \mathtt{blank}(o)$.

□

With this lemma in place, we can state and prove a lemma about the impact of compromised nodes on a certain observation. It states that the probability of making a certain observation by means of a set N of nodes can be computed by the probability that the core of this observation already suffices to make this observation, under the assumption that the remaining nodes (outside this core) make the corresponding blanked observation. Intuitively, this means that the core is sufficient for making the observation, and that the remaining nodes can only infer $\bot$ for those elements of the information that cannot be observed. This lemma leverages Lemma 4.4.2 from individual circuits to probabilities for making observations.

Lemma 4.4.3 (Impact of Compromised Nodes)**.** *Let $\mathsf{S}_a \in \mathcal{S}$ be a sender and $\mathsf{R}_b \in \mathcal{R}$ be a recipient. Let $o \in \mathsf{NatObs}$ be a natural observation, $\mathit{Core} \in \mathtt{core}(o)$, and $N \subseteq \mathcal{E}$ be a set of compromised Tor entities such that $\mathit{Core} \subseteq N$. Then we have*

$$P^{ab}[\mathcal{OC}(N, o)] = P^{ab}[\mathcal{OC}(\mathit{Core}, o) \cap \mathcal{OC}_{\mathtt{blank}}(N \setminus \mathit{Core}, o)].$$

Proof. We have

$$\begin{aligned} & P^{ab}[\mathcal{OC}(N,o)] = \texttt{Pr}\left[c \in \mathcal{OC}(N,o); \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_a, \mathsf{R}_b)\right] \\ = \ & \texttt{Pr}\left[\textsc{obs}[N](\mathsf{C}) = o; \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_a, \mathsf{R}_b)\right] \\ = \ & \texttt{Pr}\left[\textsc{obs}[\mathit{Core}](\mathsf{C}) = o \wedge \textsc{obs}[N \setminus \{\mathit{Core}\}](\mathsf{C}) = \texttt{blank}(o); \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_a, \mathsf{R}_b)\right] \\ = \ & \texttt{Pr}\left[c \in \mathcal{OC}(\mathit{Core},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \mathit{Core}, o); \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_a, \mathsf{R}_b)\right] \\ = \ & P^{ab}[\mathcal{OC}(\mathit{Core},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \mathit{Core}, o)], \end{aligned}$$

where the fourth equality follows from Lemma 4.4.2; the remaining equalities hold by definition of P^{ab} and $\mathcal{OC}$. □

For any given observation, more than the core of the observation is visible to the adversary. We now analyze the impact that compromising any such visible Tor entity may have on the impact of an observation, if the entity is not already contained in the core of the observation.

We show that such entries can be of two types: (i) entities that are superfluous, i.e., compromising them does not change the impact of the observation, and (ii) entities that are destructive, i.e., compromising them makes the observation impossible and thus completely removes the impact of an observation. We combine these insights in the following lemma, where we show that ignoring all visible entities except for the core can only increase the impact of an observation.

Lemma 4.4.4 (Impact of Visible (Non-core) Nodes)**.** *Let $\mathsf{S}_a, \mathsf{S}_c \in \mathcal{S}$ be two senders and $\mathsf{R}_b, \mathsf{R}_d \in \mathcal{R}$ be two recipients. Let $o \in \mathsf{NatObs}$ be a natural observation, $\mathit{Core} \in \texttt{core}(o)$, and $N \subseteq \mathcal{E}$ be a set of compromised Tor entities such that $\mathit{Core} \subseteq N$. Then, for $N_V = N \setminus (V(o) \setminus \mathit{Core})$,*

$$\phi\left(P^{ab}[\mathcal{OC}(N,o)], P^{cd}[\mathcal{OC}(N,o)]\right) \leq \phi\left(P^{ab}[\mathcal{OC}(N_V,o)], P^{cd}[\mathcal{OC}(N_V,o)]\right)$$

Proof. Let $\mathsf{S}_a, \mathsf{S}_c \in \mathcal{S}$ be two senders and $\mathsf{R}_b, \mathsf{R}_d \in \mathcal{R}$ be two recipients. Let $o \in \mathsf{NatObs}$ be a natural observation, $\mathit{Core} \in \texttt{core}(o)$, and $N \subseteq \mathcal{E}$ be a set of compromised Tor entities such that $\mathit{Core} \subseteq N$, and let $N_V = N \setminus (V(o) \setminus \mathit{Core})$. By Lemma 4.4.2 we know that $\mathcal{OC}(N,o) \subseteq \mathcal{OC}(\mathit{Core},o)$.

Let $x \in V(o) \setminus \mathit{Core}$ be any visible Tor entity of the observation that does not belong to the core. We show that we can remove x from N without reducing the impact. If $x \notin N$, $\phi\left(P^{ab}[\mathcal{OC}(N,o)], P^{cd}[\mathcal{OC}(N,o)]\right) = \phi\left(P^{ab}[\mathcal{OC}(N \setminus \{x\},o)], P^{cd}[\mathcal{OC}(N \setminus \{x\},o)]\right)$ trivially holds.

If $x \in N$, we distinguish two cases:

- There is a circuit $\mathsf{C} \in \mathcal{OC}(\mathit{Core},o)$ s.t. $\mathsf{C} \notin \mathcal{OC}_{\texttt{blank}}(\{x\},o)$. By our assumptions on path selection algorithms from Section 4.2.4 we know that x cannot occur in C twice with non-zero probability, so the position in which x is observed must lead to an observation that is incompatible with $\texttt{blank}(o)$. However, by definition of $\mathcal{OC}(Core, o)$ and V, for every other circuit $\mathsf{C}' \in \mathcal{OC}(Core, o)$, x is in the same position, which leads to an observation of the same form and thus $\mathsf{C}' \notin \mathcal{OC}_{\texttt{blank}}(\{x\}, o)$ must also hold. Consequently,

$$\begin{aligned}
& \phi\left(P^{ab}[\mathcal{OC}(N,o)], P^{cd}[\mathcal{OC}(N,o)]\right) \\
= \; & \phi\Big(P^{ab}[\mathcal{OC}(Core,o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus Core, o)], \\
& P^{cd}[\mathcal{OC}(Core,o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus Core, o)]\Big) \\
= \; & \phi\left(P^{ab}[\emptyset], P^{cd}[\emptyset]\right) \\
= \; & \phi(0,0) \\
\le \; & \phi\left(P^{ab}[\mathcal{OC}(N \setminus \{x\}, o)], P^{cd}[\mathcal{OC}(N \setminus \{x\}, o)]\right)
\end{aligned}$$

- Otherwise, we have $\mathcal{OC}(Core, o) \subseteq \mathcal{OC}_{\texttt{blank}}(\{x\}, o)]$ and thus,

$$\begin{aligned}
& \phi\left(P^{ab}[\mathcal{OC}(N,o)], P^{cd}[\mathcal{OC}(N,o)]\right) \\
= \; & \phi\left(P^{ab}[\mathcal{OC}(N \setminus \{x\}, o)], P^{cd}[\mathcal{OC}(N \setminus \{x\}, o)]\right).
\end{aligned}$$

Applying this reasoning for every such element $x \in V(o) \setminus Core$ concludes the proof. □

4.4.3 Proof for Sender Anonymity

We now combine the results for indirect impacts and derive our bound for sender anonymity. To this end, we first show that for any given set of compromised Tor nodes we can derive a bound according to our definitions for indirect impact.

Lemma 4.4.5 (Observations for Sender Anonymity). *Let* $\mathsf{S}_0, \mathsf{S}_1 \in \mathcal{S}$ *be two senders and* $\mathsf{R}_0 \in \mathcal{R}$ *be a recipient, and let* $N = N' \cup \{\mathsf{R}_0\}$ *be a set of compromised Tor entities with* $N' \subseteq \mathcal{N}$ *being a set of compromised Tor nodes. Then we have*

$$\begin{aligned}
& \sum_{o \in \mathcal{O}} \phi\big(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\big) \\
\le \; & \texttt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{R}_0) + \sum_{\mathsf{n} \in N'} \Bigg(\texttt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{n}) + \texttt{impact}_{\texttt{Rec1}}(\mathsf{n}) \\
& + \sum_{\mathsf{n}' \in N' \setminus \{\mathsf{n}\}} \left(\texttt{impact}^{(10),(00)}_{\texttt{indirect}}(\mathsf{n},\mathsf{n}') + \texttt{impact}_{\texttt{Rec2}}(\mathsf{n},\mathsf{n}') \right) \Bigg)
\end{aligned}$$

Proof. Let $\mathsf{S}_0, \mathsf{S}_1 \in \mathcal{S}$ be two senders and $\mathsf{R}_0 \in \mathcal{R}$ be a recipient, and let $N = N' \cup \{\mathsf{R}_0\}$ be a set of compromised Tor entities with $N' \subseteq \mathcal{N}$ being a set of compromised Tor nodes. We define the following sets of observations for sender anonymity:

- $\mathcal{O}_{g,\mathsf{SA}} := \{(n_1, \ldots, n_5) \in \mathcal{O} \text{ s.t. } n_1 \neq \perp\}$,
- $\mathcal{O}_{m,\mathsf{SA}} := \{(n_1, \ldots, n_5) \in \mathcal{O} \setminus \mathcal{O}_{g,\mathsf{SA}} \text{ s.t. } n_2 \neq \perp\}$,
- $\mathcal{O}_{x,\mathsf{SA}} := \{(n_1, \ldots, n_5) \in \mathcal{O} \setminus (\mathcal{O}_{g,\mathsf{SA}} \cup \mathcal{O}_{m,\mathsf{SA}}) \text{ s.t. } n_3 \neq \perp\}$,
- $\mathcal{O}_{R,\mathsf{SA}} := \mathcal{O} \setminus (\mathcal{O}_{g,\mathsf{SA}} \cup \mathcal{O}_{m,\mathsf{SA}} \cup \mathcal{O}_{x,\mathsf{SA}})$.

As these sets define a partitioning of $\mathcal{O}$, we get

$$\begin{aligned}
&\sum_{o\in\mathcal{O}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right)\\
=&\sum_{o\in\mathcal{O}_{g,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) + \sum_{o\in\mathcal{O}_{m,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right)\\
+&\sum_{o\in\mathcal{O}_{x,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) + \sum_{o\in\mathcal{O}_{R,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right)
\end{aligned}$$

We now consider each of the sets of observations individually.

- $\mathcal{O}_{g,\mathsf{SA}}$. We first show that (in addition to R_0) a compromised guard node is required and sufficient for making an observation in $\mathcal{O}_{g,\mathsf{SA}}$ and that additionally compromised nodes do not contribute to the impact of such observations. Let $\mathsf{C} = (\mathsf{S}_a, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_0) \in \mathsf{Circuits}$ be any circuit with $\mathsf{S}_a \in \{\mathsf{S}_0, \mathsf{S}_1\}$ s.t. $o = \textsc{obs}[N](\mathsf{C}) \in \mathcal{O}_{g,\mathsf{SA}}$ is the observation made by the adversary for this circuit. By definition of $\mathcal{O}_{g,\mathsf{SA}}$, S_a is part of the observation. Since $\{\mathsf{S}_0, \mathsf{S}_1\} \cap N = \emptyset$, we know that $\mathsf{n_g} \in N$ is required to make the observation. Since $\mathsf{n_g} \in N$ is also sufficient to observe the sender we get $P^{00}[\mathcal{OC}(N,o)] = 0$ or $P^{10}[\mathcal{OC}(N,o)] = 0$. By definition of ϕ we know that for these observations $\phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) = P^{00}[\mathcal{OC}(N,o)]$. Thus,

$$\begin{aligned}
&\sum_{o\in\mathcal{O}_{g,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right)\\
=&\sum_{o\in\mathcal{O}_{g,\mathsf{SA}}} P^{00}[\mathcal{OC}(N,o)]\\
=&\sum_{\substack{\mathsf{n_g}\in N'\\ \mathsf{n_m},\mathsf{n_x}\in\mathcal{N}}} \mathtt{Pr}\left[\mathsf{C} = (\mathsf{S}_0, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_0), \mathsf{C} \leftarrow \textsc{ps}(\mathsf{S}_0, \mathsf{R}_0)\right]\\
=&\sum_{\mathsf{n_g}\in N'} \sum_{o\in\mathcal{O}_{g,\mathsf{SA}}} \phi\big(P^{00}[\mathcal{OC}(\{\mathsf{n_g}\}, o)], P^{10}[\mathcal{OC}(\{\mathsf{n_g}\}, o)]\big).
\end{aligned}$$

- $\mathcal{O}_{m,\mathsf{SA}}$. We first show that (in addition to R_0) a compromised middle node is required and sufficient for making an observation in $\mathcal{O}_{m,\mathsf{SA}}$. We then argue that we can remove the recipient from the computation without modifying the probability and, finally, that additionally compromised nodes do not contribute to the impact of such observations. Let $\mathsf{C} = (\mathsf{S}_a, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_0) \in \mathsf{Circuits}$ be any circuit with $\mathsf{S}_a \in \{\mathsf{S}_0, \mathsf{S}_1\}$ s.t. $o = \textsc{obs}[N](\mathsf{C}) \in \mathcal{O}_{m,\mathsf{SA}}$ is the observation made by the adversary for this circuit. We have $\mathtt{core}(o) = \{\{\mathsf{n_m}, \mathsf{n_x}\}, \{\mathsf{n_m}, \mathsf{R}_0\}\}$ and set $\mathit{Core} = \{\mathsf{n_m}, \mathsf{R}_0\}$. By Lemma 4.4.4 we know that

$$\begin{aligned} & \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) \\ \leq\ & \phi\left(P^{00}[\mathcal{OC}(N \setminus \{\mathsf{n_g}, \mathsf{n_x}\}, o)], P^{10}[\mathcal{OC}(N \setminus \{\mathsf{n_g}, \mathsf{n_x}\}, o)]\right) \end{aligned}$$

All nodes of the circuit are visible in the observation. Since all nodes in $N \setminus V(o)$ can thus never lead to the observation o and consequently, compromising them does not modify the probability of the observation, we know that

$$\begin{aligned} & \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) \\ \leq\ & \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_m}, \mathsf{R}_0\}, o)], P^{10}[\mathcal{OC}(\{\mathsf{n_m}, \mathsf{R}_0\}, o)]\right) \end{aligned}$$

Moreover, we know that

$$\textsc{obs}[\{\mathsf{n_m}, \mathsf{R}_0\}](\mathsf{C}) = (\bot, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_0),$$

and $\textsc{obs}[\{\mathsf{n_m}\}](\mathsf{C}) = (\bot, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \bot) =: o'$. Since there is only one recipient and since this holds for all respective circuits,

$$P^{a0}[\mathcal{OC}(\{\mathsf{n_m}, \mathsf{R}_0\}, o)] = P^{a0}[\mathcal{OC}(\{\mathsf{n_m}\}, o')],$$

and, consequently,

$$\begin{aligned} & \sum_{o \in \mathcal{O}_{m,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) \\ \leq\ & \sum_{o \in \mathcal{O}_{m,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_m}\}, o)], P^{10}[\mathcal{OC}(\{\mathsf{n_m}\}, o)]\right) \end{aligned}$$

- $\mathcal{O}_{x,\mathsf{SA}}$. All observations $o \in \mathcal{O}_{x,\mathsf{SA}}$ are of the form $(\bot, \bot, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_0)$ and we have $\mathtt{core}(o) = \{\{\mathsf{n_x}\}\}$ and $V(o) = \{\mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_0\}$. By Lemma 4.4.4 we know that

$$\begin{aligned} & \sum_{o \in \mathcal{O}_{x,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) \\ \leq\ & \sum_{o \in \mathcal{O}_{x,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N \setminus \{\mathsf{n_m}, \mathsf{R}_0\}, o)], P^{10}[\mathcal{OC}(N \setminus \{\mathsf{n_m}, \mathsf{R}_0\}, o)]\right). \end{aligned}$$

If $\mathsf{n_x} \notin N$ then

$$\begin{aligned} & \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) \\ =\ & \phi(0,0) = 0. \end{aligned}$$

Otherwise, we rewrite the probability according to Lemma 4.4.3 and get

$$\phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right)$$

$$\begin{aligned}
&\leq \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_x}\}, o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus V(o), o)],\right.\\
&\quad \left. P^{10}[\mathcal{OC}(\{\mathsf{n_x}\}, o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus V(o), o)]\right)\\
&= \phi\left(\sum_{\mathsf{n_g} \in \mathcal{N}\setminus N'} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g} \in \mathcal{N}\setminus N'} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
&= \phi\left(\sum_{\mathsf{n_g} \in \mathcal{N}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_g} \in N'} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g} \in \mathcal{N}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_g} \in N'} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
&\leq \phi\left(\sum_{\mathsf{n_g} \in \mathcal{N}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g} \in \mathcal{N}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) + \phi\left(\sum_{\mathsf{n_g} \in N'} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g} \in N'} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
&\leq \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_x}\}, o)], P^{10}[\mathcal{OC}(\{\mathsf{n_x}\}, o)]\right) + \phi\left(\sum_{\mathsf{n_g} \in N'} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g} \in N'} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right).
\end{aligned}$$

Thus, if we sum over all observations $o \in \mathcal{O}_{x,\mathsf{SA}}$ we get

$$\begin{aligned}
&\sum_{o \in \mathcal{O}_{x,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N, o)], P^{10}[\mathcal{OC}(N, o)]\right)\\
\leq &\sum_{o \in \mathcal{O}_{x,\mathsf{SA}}} \left(\phi\left(P^{00}[\mathcal{OC}(\{o.n_4\}, o)], P^{10}[\mathcal{OC}(\{o.n_4\}, o)]\right)\right.\\
&\left. + \phi\left(\sum_{\mathsf{n_g} \in N'} P^{1,0}_{\mathsf{n_g},o.n_3,o.n_4}, \sum_{\mathsf{n_g} \in N'} P^{0,0}_{\mathsf{n_g},o.n_3,o.n_4}\right)\right)\\
= &\sum_{\mathsf{n_x} \in N'} \sum_{o \in \mathcal{O}_{x,\mathsf{SA}}} \left(\phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_x}\}, o)], P^{10}[\mathcal{OC}(\{\mathsf{n_x}\}, o)]\right)\right)\\
&+ \sum_{\mathsf{n_x} \in N'} \sum_{\mathsf{n_m} \in \mathcal{N}} \phi\left(\sum_{\mathsf{n_g} \in N'} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g} \in N'} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
\leq &\sum_{\mathsf{n_x} \in N'} \sum_{o \in \mathcal{O}_{x,\mathsf{SA}}} \left(\phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_x}\}, o)], P^{10}[\mathcal{OC}(\{\mathsf{n_x}\}, o)]\right)\right)\\
&+ \sum_{\substack{\mathsf{n_g} \in N' \\ \mathsf{n_x} \in N'}} \texttt{impact}^{(10),(00)}_{\texttt{indirect}}(\mathsf{n_g}, \mathsf{n_x}).
\end{aligned}$$

- $\mathcal{O}_{R,\mathsf{SA}}$. Let $o = (\bot, \bot, \bot, \mathsf{n_x}, \mathsf{R_0})$ be any observation in $\mathcal{O}_{R,\mathsf{SA}}$. Since $\texttt{core}(o) = \{\{\mathsf{R_0}\}\}$ we have *Core* $= \{\mathsf{R_0}\}$. By Lemma 4.4.4 we know that

$$\begin{aligned}
&\phi\left(P^{00}[\mathcal{OC}(N, o)], P^{10}[\mathcal{OC}(N, o)]\right)\\
\leq\ &\phi\left(P^{00}[\mathcal{OC}(N \setminus \{\mathsf{n_x}\}, o)], P^{10}[\mathcal{OC}(N \setminus \{\mathsf{n_x}\}, o)]\right)
\end{aligned}$$

We rewrite the probability according to Lemma 4.4.3 and yield

$$\phi\left(P^{00}[\mathcal{OC}(N \setminus \{\mathsf{n_x}\}, o)], P^{10}[\mathcal{OC}(N \setminus \{\mathsf{n_x}\}, o)]\right)$$

$$
\begin{aligned}
&= \phi\left(P^{00}[\mathcal{OC}(Core, o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_x}, \mathsf{R_0}\}, o)],\right.\\
&\qquad \left.P^{10}[\mathcal{OC}(Core, o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_x}, \mathsf{R_0}\}, o)]\right)\\
&= \phi\left(\sum_{\substack{\mathsf{n_g} \in \mathcal{N}\setminus N' \\ \mathsf{n_m} \in \mathcal{N}\setminus N'}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\substack{\mathsf{n_g} \in \mathcal{N}\setminus N' \\ \mathsf{n_m} \in \mathcal{N}\setminus N'}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}},\right)\\
&= \phi\left(\sum_{\substack{\mathsf{n_g} \in \mathcal{N} \\ \mathsf{n_m} \in \mathcal{N}}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\substack{\mathsf{n} \in N' \\ \mathsf{n'} \in \mathcal{N}}} \left(P^{0,0}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}} + P^{0,0}_{\mathsf{n'},\mathsf{n},\mathsf{n_x}}\right) + \sum_{\substack{\mathsf{n_g} \in N' \\ \mathsf{n_m} \in N'}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}},\right.\\
&\qquad \left.\sum_{\substack{\mathsf{n_g} \in \mathcal{N} \\ \mathsf{n_m} \in \mathcal{N}}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\substack{\mathsf{n} \in N' \\ \mathsf{n'} \in \mathcal{N}}} \left(P^{1,0}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}} + P^{1,0}_{\mathsf{n'},\mathsf{n},\mathsf{n_x}}\right) + \sum_{\substack{\mathsf{n_g} \in N' \\ \mathsf{n_m} \in N'}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
&\leq \phi\left(\sum_{\substack{\mathsf{n_g} \in \mathcal{N} \\ \mathsf{n_m} \in \mathcal{N}}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\substack{\mathsf{n_g} \in \mathcal{N} \\ \mathsf{n_m} \in \mathcal{N}}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
&\qquad +\phi\left(\sum_{\substack{\mathsf{n} \in N' \\ \mathsf{n'} \in \mathcal{N}}} \left(P^{1,0}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}} + P^{1,0}_{\mathsf{n'},\mathsf{n},\mathsf{n_x}}\right), \sum_{\substack{\mathsf{n} \in N' \\ \mathsf{n'} \in \mathcal{N}}} \left(P^{0,0}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}} + P^{0,0}_{\mathsf{n'},\mathsf{n},\mathsf{n_x}}\right)\right)\\
&\qquad +\phi\left(\sum_{\substack{\mathsf{n_g} \in N' \\ \mathsf{n_m} \in N'}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\substack{\mathsf{n_g} \in N' \\ \mathsf{n_m} \in N'}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
&\leq \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{R_0}\}, o)], P^{10}[\mathcal{OC}(\{\mathsf{R_0}\}, o)]\right)\\
&\qquad + \sum_{\mathsf{n} \in N'} \phi\left(\sum_{\mathsf{n'} \in \mathcal{N}} \left(P^{1,0}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}} + P^{1,0}_{\mathsf{n'},\mathsf{n},\mathsf{n_x}}\right), \sum_{\mathsf{n'} \in \mathcal{N}} \left(P^{0,0}_{\mathsf{n},\mathsf{n'},\mathsf{n_x}} + P^{0,0}_{\mathsf{n'},\mathsf{n},\mathsf{n_x}}\right)\right)\\
&\qquad + \sum_{\substack{\mathsf{n_g} \in N' \\ \mathsf{n_m} \in N'}} \phi\left(P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)
\end{aligned}
$$

We combine these individual bounds for the sets of observations and yield

$$
\begin{aligned}
&\sum_{o \in \mathcal{O}} \phi\left(P^{00}[\mathcal{OC}(N, o)], P^{10}[\mathcal{OC}(N, o)]\right)\\
= &\sum_{o \in \mathcal{O}_{g,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N, o)], P^{10}[\mathcal{OC}(N, o)]\right)\\
+ &\sum_{o \in \mathcal{O}_{m,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N, o)], P^{10}[\mathcal{OC}(N, o)]\right)
\end{aligned}
$$

$$
\begin{aligned}
+ & \sum_{o \in \mathcal{O}_{x,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) \\
+ & \sum_{o \in \mathcal{O}_{R,\mathsf{SA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\right) \\
\leq & \sum_{o \in \mathcal{O}_{g,\mathsf{SA}}} \sum_{\mathsf{n_g} \in N'} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_g}\},o)], P^{10}[\mathcal{OC}(\{\mathsf{n_g}\},o)]\right) \\
+ & \sum_{o \in \mathcal{O}_{m,\mathsf{SA}}} \sum_{\mathsf{n_m} \in N'} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_m}\},o)], P^{10}[\mathcal{OC}(\{\mathsf{n_m}\},o)]\right) \\
+ & \sum_{o \in \mathcal{O}_{x,\mathsf{SA}}} \sum_{\mathsf{n_x} \in N'} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_x}\},o)], P^{10}[\mathcal{OC}(\{\mathsf{n_x}\},o)]\right) \\
+ & \sum_{\substack{\mathsf{n_g} \in N' \\ \mathsf{n_x} \in N'}} \left(\mathtt{impact}^{(10),(00)}_{\mathtt{indirect}}\left(\mathsf{n_g},\mathsf{n_x}\right)\right) \\
+ & \sum_{\substack{o \in \mathcal{O}_{R,\mathsf{SA}} \\ (n_1,n_2,n_3,n_4,n_5):=o}} \Bigg(\phi\left(P^{00}[\mathcal{OC}(\{\mathsf{R}_0\},o)], P^{10}[\mathcal{OC}(\{\mathsf{R}_0\},o)]\right) \\
& + \sum_{\mathsf{n} \in N'} \phi\left(\sum_{\mathsf{n}' \in \mathcal{N}} \left(P^{1,0}_{\mathsf{n},\mathsf{n}',n_4} + P^{1,0}_{\mathsf{n}',\mathsf{n},n_4}\right), \sum_{\mathsf{n}' \in \mathcal{N}} \left(P^{0,0}_{\mathsf{n},\mathsf{n}',n_4} + P^{0,0}_{\mathsf{n}',\mathsf{n},n_4}\right)\right) \\
& + \sum_{\substack{\mathsf{n_g} \in N' \\ \mathsf{n_m} \in N'}} \phi\left(P^{0,0}_{\mathsf{n_g},\mathsf{n_m},n_4}, P^{1,0}_{\mathsf{n_g},\mathsf{n_m},n_4}\right)\Bigg) \\
\leq & \sum_{\mathsf{n} \in N'} \Bigg(\sum_{o \in \mathsf{Obs}_{\not\perp}} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n}\},o)], P^{10}[\mathcal{OC}(\{\mathsf{n}\},o)]\right) + \sum_{\mathsf{n}' \in N'} \mathtt{impact}^{(10),(00)}_{\mathtt{indirect}}\left(\mathsf{n},\mathsf{n}'\right) \\
+ & \mathtt{impact}_{\mathtt{Rec1}}(\mathsf{n}) + \sum_{\mathsf{n}' \in N'} \mathtt{impact}_{\mathtt{Rec2}}(\mathsf{n},\mathsf{n}')\Bigg) \\
+ & \sum_{o \in \mathsf{Obs}_{\not\perp}} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{R}_0\},o)], P^{10}[\mathcal{OC}(\{\mathsf{R}_0\},o)]\right) \\
= & \mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}\left(\mathsf{R}_0\right) + \sum_{\mathsf{n} \in N'} \Big(\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}\left(\mathsf{n}\right) + \mathtt{impact}_{\mathtt{Rec1}}(\mathsf{n}) \\
+ & \sum_{\mathsf{n}' \in N' \setminus \{\mathsf{n}\}} \left(\mathtt{impact}^{(10),(00)}_{\mathtt{indirect}}\left(\mathsf{n},\mathsf{n}'\right) + \mathtt{impact}_{\mathtt{Rec2}}(\mathsf{n},\mathsf{n}')\right)\Big)
\end{aligned}
$$

The last equation holds because for all $\mathsf{n} \in \mathcal{N}$, $\mathtt{impact}^{(ab),(cd)}_{\mathtt{indirect}}\left(\mathsf{n},\mathsf{n}\right) = 0$, since all probabilities of circuits are zero if the same node is used more than once in the same circuit. $\square$

4.4.4 Proof for Recipient Anonymity

We now derive our bound for recipient anonymity. Both the lemma and the proof are completely analogous to the case of sender anonymity.

Lemma 4.4.6 (Observations for Recipient Anonymity). *Let $\mathsf{S}_0 \in \mathcal{S}$ be a sender and $\mathsf{R}_0, \mathsf{R}_1 \in \mathcal{R}$ be two recipients, and let $N = N' \cup \{\mathsf{S}_0\}$ be a set of compromised Tor entities with $N' \subseteq \mathcal{N}$ being a set of compromised Tor nodes. Then we have*

$$\begin{aligned}
&\sum_{o\in\mathcal{O}} \phi\big(P^{00}[\mathcal{OC}(N,o)], P^{01}[\mathcal{OC}(N,o)]\big)\\
\leq\ & \mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{S}_0) + \sum_{\mathsf{n}\in N'} \Bigg(\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n}) + \mathtt{impact}_{\mathtt{Sen1}}(\mathsf{n})\\
+\ & \sum_{\mathsf{n}'\in N'\setminus\{\mathsf{n}\}} \Big(\mathtt{impact}^{(01),(00)}_{\mathtt{indirect}}(\mathsf{n},\mathsf{n}') + \mathtt{impact}_{\mathtt{Sen2}}(\mathsf{n},\mathsf{n}')\Big)\Bigg)
\end{aligned}$$

Proof. Let $\mathsf{S}_0 \in \mathcal{S}$ be a sender and $\mathsf{R}_0, \mathsf{R}_1 \in \mathcal{R}$ be two recipients, and let $N = N' \cup \{\mathsf{S}_0\}$ be a set of compromised Tor entities with $N' \subseteq \mathcal{N}$ being a set of compromised Tor nodes. We define the following sets of observations for recipient anonymity:

- $\mathcal{O}_{x,\mathsf{RA}} := \{(n_1, \ldots, n_5) \in \mathcal{O}$ s.t. $n_5 \neq \bot\}$,
- $\mathcal{O}_{m,\mathsf{RA}} := \{(n_1, \ldots, n_5) \in \mathcal{O} \setminus \mathcal{O}_{x,\mathsf{RA}}$ s.t. $n_4 \neq \bot\}$,
- $\mathcal{O}_{g,\mathsf{RA}} := \{(n_1, \ldots, n_5) \in \mathcal{O} \setminus (\mathcal{O}_{x,\mathsf{RA}} \cup \mathcal{O}_{m,\mathsf{RA}})$ s.t. $n_3 \neq \bot\}$,
- $\mathcal{O}_{S,\mathsf{RA}} := \mathcal{O} \setminus (\mathcal{O}_{x,\mathsf{RA}} \cup \mathcal{O}_{m,\mathsf{RA}} \cup \mathcal{O}_{g,\mathsf{RA}})$.

As these sets define a partitioning of $\mathcal{O}$, we get

$$\begin{aligned}
&\sum_{o\in\mathcal{O}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{01}[\mathcal{OC}(N,o)]\right)\\
=\ & \sum_{o\in\mathcal{O}_{x,\mathsf{RA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{01}[\mathcal{OC}(N,o)]\right)\\
+\ & \sum_{o\in\mathcal{O}_{m,\mathsf{RA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{01}[\mathcal{OC}(N,o)]\right)\\
+\ & \sum_{o\in\mathcal{O}_{g,\mathsf{RA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{01}[\mathcal{OC}(N,o)]\right)\\
+\ & \sum_{o\in\mathcal{O}_{S,\mathsf{RA}}} \phi\left(P^{00}[\mathcal{OC}(N,o)], P^{01}[\mathcal{OC}(N,o)]\right)
\end{aligned}$$

The rest of the proof is completely analogous to the proof for Lemma 4.4.5. To give some intuition into the proof, we describe which parts of this proof correspond to which parts of the proof for Lemma 4.4.5.

- $\mathcal{O}_{x,\mathsf{RA}}$ corresponds to $\mathcal{O}_{g,\mathsf{SA}}$ in Lemma 4.4.5.
- $\mathcal{O}_{m,\mathsf{RA}}$ corresponds to $\mathcal{O}_{m,\mathsf{SA}}$ in Lemma 4.4.5.
- $\mathcal{O}_{g,\mathsf{RA}}$ corresponds to $\mathcal{O}_{x,\mathsf{SA}}$ in Lemma 4.4.5.
- $\mathcal{O}_{S,\mathsf{RA}}$ corresponds to $\mathcal{O}_{R,\mathsf{SA}}$ in Lemma 4.4.5.

We combine these bounds and yield

$$\begin{aligned}
& \mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(N) \\
= & \sum_{o\in\mathcal{O}} \phi\big(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\big) \\
= & \sum_{o\in\mathcal{O}_{x,\mathsf{RA}}} \phi\big(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\big) \\
+ & \sum_{o\in\mathcal{O}_{m,\mathsf{RA}}} \phi\big(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\big) \\
+ & \sum_{o\in\mathcal{O}_{g,\mathsf{RA}}} \phi\big(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\big) \\
+ & \sum_{o\in\mathcal{O}_{S,\mathsf{RA}}} \phi\big(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\big) \\
\leq & \sum_{o\in\mathcal{O}_{x,\mathsf{RA}}} \sum_{\mathsf{n_x}\in N'} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_x}\},o)], P^{01}[\mathcal{OC}(\{\mathsf{n_x}\},o)]\right) \\
+ & \sum_{o\in\mathcal{O}_{m,\mathsf{RA}}} \sum_{\mathsf{n_m}\in N'} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_m}\},o)], P^{01}[\mathcal{OC}(\{\mathsf{n_m}\},o)]\right) \\
+ & \sum_{o\in\mathcal{O}_{g,\mathsf{RA}}} \left(\sum_{\mathsf{n_g}\in N'} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{n_g}\},o)], P^{01}[\mathcal{OC}(\{\mathsf{n_g}\},o)]\right)\right) \\
+ & \sum_{\substack{\mathsf{n_g}\in N' \\ \mathsf{n_x}\in N'}} \left(\mathtt{impact}^{(01),(00)}_{\mathtt{indirect}}(\mathsf{n_g},\mathsf{n_x})\right) \\
+ & \sum_{\substack{o\in\mathcal{O}_{S,\mathsf{RA}} \\ (n_1,n_2,n_3,n_4,n_5):=o}} \Bigg(\phi\left(P^{00}[\mathcal{OC}(\{\mathsf{S_0}\},o)], P^{01}[\mathcal{OC}(\{\mathsf{S_0}\},o)]\right) \\
& + \sum_{\mathsf{n}\in N'} \phi\left(\sum_{\mathsf{n'}\in\mathcal{N}}\left(P^{0,1}_{n_2,\mathsf{n},\mathsf{n'}} + P^{0,1}_{n_2,\mathsf{n'},\mathsf{n}}\right), \sum_{\mathsf{n'}\in\mathcal{N}}\left(P^{0,0}_{n_2,\mathsf{n},\mathsf{n'}} + P^{0,0}_{n_2,\mathsf{n'},\mathsf{n}}\right)\right) \\
& + \sum_{\substack{\mathsf{n_m}\in N' \\ \mathsf{n_x}\in N'}} \phi\left(P^{0,0}_{n_2,\mathsf{n_m},\mathsf{n_x}}, P^{0,1}_{n_2,\mathsf{n_m},\mathsf{n_x}}\right)\Bigg) \\
\leq & \sum_{\mathsf{n}\in N'} \Bigg(\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n}) + \sum_{\mathsf{n'}\in N'} \mathtt{impact}^{(01),(00)}_{\mathtt{indirect}}\left(\mathsf{n},\mathsf{n'}\right)
\end{aligned}$$

$$
\begin{aligned}
&+ \quad \texttt{impact}_{\texttt{Sen1}}(\mathsf{n}) + \sum_{\mathsf{n}' \in N'} \texttt{impact}_{\texttt{Sen2}}(\mathsf{n}, \mathsf{n}')\Bigg) \\
&+ \sum_{o \in \mathsf{Obs}_{\not\perp}} \phi\left(P^{00}[\mathcal{OC}(\{\mathsf{S}_0\}, o)], P^{01}[\mathcal{OC}(\{\mathsf{S}_0\}, o)]\right) \\
&= \texttt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{S}_0) + \sum_{\mathsf{n} \in N'} \Bigg(\texttt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n}) + \texttt{impact}_{\texttt{Sen1}}(\mathsf{n}) \\
&+ \sum_{\mathsf{n}' \in N' \setminus \{\mathsf{n}\}} \left(\texttt{impact}^{(01),(00)}_{\texttt{indirect}}\left(\mathsf{n}, \mathsf{n}'\right) + \texttt{impact}_{\texttt{Sen2}}(\mathsf{n}, \mathsf{n}')\right)\Bigg)
\end{aligned}
$$

The last equation holds because for all $\mathsf{n} \in \mathcal{N}$, $\texttt{impact}^{(ab),(cd)}_{\texttt{indirect}}(\mathsf{n}, \mathsf{n}) = 0$, since all probabilities of circuits are zero if the same node is used more than once in the same circuit. $\square$

4.4.5 Proof for Relationship Anonymity

We combine the results for indirect impacts and derive our bound for relationship anonymity. In contrast to sender anonymity and relationship anonymity, two aspects are different for relationship anonymity: (i) for relationship anonymity, N contains neither a sender nor a recipient from the challenge message, and (ii) for relationship anonymity we have to consider four distributions over circuits instead of two distributions over circuits. Recall that we write $P^{ab,cd}[C]$ instead of $\frac{1}{2}\left(P^{ab}[C] + P^{cd}[C]\right)$.

Lemma 4.4.7 (Observations for Relationship Anonymity). *Let* $\mathsf{S}_0, \mathsf{S}_1$ *be two senders and* $\mathsf{R}_0, \mathsf{R}_1$ *be two recipients, and let* $N \subseteq \mathcal{N}$ *be a set of compromised Tor nodes. Under the assumption that* $P^{00,11}[\mathcal{OC}(N, (\perp, \perp, \perp, \perp, \perp))] \leq P^{01,10}[\mathcal{OC}(N, (\perp, \perp, \perp, \perp, \perp))]$

$$
\begin{aligned}
&\sum_{o \in \mathcal{O}} \phi\left(P^{00,11}[\mathcal{OC}(N, o)], P^{01,10}[\mathcal{OC}(N, o)]\right) \\
\leq &\sum_{\mathsf{n} \in N} \Bigg(\texttt{impact}^{\mathcal{O}}_{\mathsf{REL}}(\{\mathsf{n}\}) + \sum_{\mathsf{n}' \in N} \left(\texttt{impact}^{\mathcal{O}\text{-}2}_{\mathsf{REL}}(\{\mathsf{n}, \mathsf{n}'\})\right) \\
&+ \sum_{\mathsf{n}' \in N} \Bigg(+\frac{1}{2}\texttt{impact}^{(01),(00)}_{\texttt{indirect}}\left(\mathsf{n}, \mathsf{n}'\right) + \frac{1}{2}\texttt{impact}^{(10),(11)}_{\texttt{indirect}}\left(\mathsf{n}, \mathsf{n}'\right) \\
&+\frac{1}{2}\texttt{impact}^{(10),(00)}_{\texttt{indirect}}\left(\mathsf{n}', \mathsf{n}\right) + \frac{1}{2}\texttt{impact}^{(01),(11)}_{\texttt{indirect}}\left(\mathsf{n}', \mathsf{n}\right)\Bigg)\Bigg)
\end{aligned}
$$

Proof. We define the following sets of observations for relationship anonymity, where each set is indexed by the positions of the compromised nodes that make the respective observations:

- $\mathcal{O}_{gx,\mathsf{REL}} := \{(n_1, \ldots, n_5) \in \mathcal{O} \text{ s.t. } (n_1 \neq \perp \wedge n_5 \neq \perp)\}$,

- $\mathcal{O}_{gm,\mathsf{REL}} := \{(n_1,\dots,n_5) \in \mathcal{O} \setminus \mathcal{O}_{gx,\mathsf{REL}} \text{ s.t. } n_1 \neq \bot \wedge n_4 \neq \bot\}$,
- $\mathcal{O}_{mx,\mathsf{REL}} := \{(n_1,\dots,n_5) \in \mathcal{O} \setminus \mathcal{O}_{gx,\mathsf{REL}} \text{ s.t. } n_2 \neq \bot \wedge n_5 \neq \bot\}$,
- $\mathcal{O}_{m,\mathsf{REL}} := \{(n_1,\dots,n_5) \in \mathcal{O} \setminus (\mathcal{O}_{gx,\mathsf{REL}} \cup \mathcal{O}_{gm,\mathsf{REL}} \cup \mathcal{O}_{mx,\mathsf{REL}}) \text{ s.t. } n_2 \neq \bot \wedge n_4 \neq \bot\}$,
- $\mathcal{O}_{g,\mathsf{REL}} := \{(n_1,\dots,n_5) \in \mathcal{O} \setminus (\mathcal{O}_{gx,\mathsf{REL}} \cup \mathcal{O}_{gm,\mathsf{REL}} \cup \mathcal{O}_{mx,\mathsf{REL}} \cup \mathcal{O}_{m,\mathsf{REL}}) \text{ s.t. } n_2 \neq \bot\}$,
- $\mathcal{O}_{x,\mathsf{REL}} := \{(n_1,\dots,n_5) \in \mathcal{O} \setminus (\mathcal{O}_{gx,\mathsf{REL}} \cup \mathcal{O}_{gm,\mathsf{REL}} \cup \mathcal{O}_{mx,\mathsf{REL}} \cup \mathcal{O}_{m,\mathsf{REL}}) \text{ s.t. } n_4 \neq \bot\}$,
- $\mathcal{O}_{r,\mathsf{REL}} := \mathcal{O} \setminus (\mathcal{O}_{gx,\mathsf{REL}} \cup \mathcal{O}_{gm,\mathsf{REL}} \cup \mathcal{O}_{mx,\mathsf{REL}} \cup \mathcal{O}_{m,\mathsf{REL}} \cup \mathcal{O}_{g,\mathsf{REL}} \cup \mathcal{O}_{x,\mathsf{REL}})$. Consequently, $\mathcal{O}_{r,\mathsf{REL}} = \{(\bot,\bot,\bot,\bot,\bot)\}$.

We now consider each of the sets of observations individually.

- $\mathcal{O}_{gx,\mathsf{REL}}$. Since neither the sender nor the recipient can be compromised for relationship anonymity, each observation in this set is of the form $o = (\mathsf{S}_a, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_b)$ and we either have $\mathsf{n_g}, \mathsf{n_x} \in N$, or $\phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) = \phi(0,0) = 0$. Since $\{\mathsf{n_g}, \mathsf{n_x}\} \in \mathtt{core}(o)$, by Lemma 4.4.4 and by the fact that the observation can only be made by the nodes $\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}$ that belong to the circuit we get
$$\begin{aligned} & \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) \\ \leq\ & \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}, \mathsf{n_x}\}, o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}, \mathsf{n_x}\}, o)]\right). \end{aligned}$$
- $\mathcal{O}_{gm,\mathsf{REL}}$. Each observation in this set is of the form $o = (\mathsf{S}_a, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \bot)$ and we have $\mathtt{core}(o) = \{\{\mathsf{S}_a, \mathsf{n_m}\}, \{\mathsf{n_g}, \mathsf{n_m}\}\}$. Since $\mathsf{S}_a \notin N$ for relationship anonymity, we assume $\mathit{Core} = \{\mathsf{n_g}, \mathsf{n_m}\} \subseteq N$ to make this observation, as otherwise $\phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) = \phi(0,0) = 0$. Since $\{\mathsf{n_g}, \mathsf{n_m}\} \in \mathtt{core}(o)$, by Lemma 4.4.4 and by the fact that the observation can only be made by the nodes $\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}$ that belong to the circuit we get
$$\begin{aligned} & \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) \\ \leq\ & \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}, \mathsf{n_m}\}, o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}, \mathsf{n_m}\}, o)]\right). \end{aligned}$$
- $\mathcal{O}_{mx,\mathsf{REL}}$. Analogously to the previous case, each observation in this set is of the form $o = (\bot, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \mathsf{R}_b)$ and we have $\mathtt{core}(o) = \{\{\mathsf{n_m}, \mathsf{n_x}\}, \{\mathsf{n_m}, \mathsf{R}_b\}\}$. Since $\mathsf{R}_b \notin N$ for relationship anonymity, we assume $\mathit{Core} = \{\mathsf{n_m}, \mathsf{n_x}\} \subseteq N$ to make this observation, as otherwise $\phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) = \phi(0,0) = 0$. Since $\{\mathsf{n_m}, \mathsf{n_x}\} \in \mathtt{core}(o)$, by Lemma 4.4.4 and by the fact that the observation can only be made by the nodes $\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}$ that belong to the circuit we get
$$\begin{aligned} & \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) \\ \leq\ & \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_m}, \mathsf{n_x}\}, o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_m}, \mathsf{n_x}\}, o)]\right). \end{aligned}$$

- $\mathcal{O}_m$. Each observation in this set is of the form $o = (\bot, \mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}, \bot)$ and we have $\texttt{core}(o) = \{\{\mathsf{n_m}\}\}$. We assume $\mathit{Core} = \{\mathsf{n_m}\} \subseteq N$ to make this observation, as otherwise $\phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) = \phi(0,0) = 0$. Since $\{\mathsf{n_m}\} \in \texttt{core}(o)$, by Lemma 4.4.4 and by the fact that the observation can only be made by the nodes $\mathsf{n_g}, \mathsf{n_m}, \mathsf{n_x}$ that belong to the circuit we get

$$\begin{aligned} & \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) \\ \leq\ & \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_m}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_m}\},o)]\right). \end{aligned}$$

- $\mathcal{O}_{g,\mathsf{REL}}$. Each observation in this set is of the form $o = (\mathsf{S}_a, \mathsf{n_g}, \mathsf{n_m}, \bot, \bot)$ and we have $\texttt{core}(o) = \{\{\mathsf{n_g}\}\}$. We assume $\mathit{Core} = \{\mathsf{n_g}\} \subseteq N$ to make this observation, as otherwise $\phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) = \phi(0,0) = 0$. Since $\{\mathsf{n_g}\} \in \texttt{core}(o)$, by Lemmas 4.4.3 and 4.4.4 we get

$$\begin{aligned} & \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) \\ \leq\ & \phi\left(P^{00,11}[\mathcal{OC}(N \setminus \{\mathsf{n_m}\},o)], P^{01,10}[\mathcal{OC}(N \setminus \{\mathsf{n_m}\},o)]\right) \\ =\ & \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_g},\mathsf{n_m}\},o)],\right. \\ & \left.P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_g},\mathsf{n_m}\},o)]\right) \end{aligned}$$

We now distinguish two cases depending on the sender S_a:

- Case $\mathsf{S}_a = \mathsf{S}_0$. Then

$$\begin{aligned} & \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_g},\mathsf{n_m}\},o)],\right. \\ & \left.P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_g},\mathsf{n_m}\},o)]\right) \\ =\ & \frac{1}{2}\phi\left(\sum_{\mathsf{n_x}\in\mathcal{N}\setminus N} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_x}\in\mathcal{N}\setminus N} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \\ =\ & \frac{1}{2}\phi\left(\sum_{\mathsf{n_x}\in\mathcal{N}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_x}\in N} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_x}\in\mathcal{N}} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_x}\in N} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \\ \leq\ & \frac{1}{2}\phi\left(\sum_{\mathsf{n_x}\in\mathcal{N}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_x}\in\mathcal{N}} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) + \frac{1}{2}\phi\left(\sum_{\mathsf{n_x}\in N} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_x}\in N} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \\ \leq\ & \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}\},o)]\right) \\ +\ & \frac{1}{2}\sum_{\mathsf{n_x}\in N}\phi\left(P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \end{aligned}$$

- Case $\mathsf{S}_a = \mathsf{S}_1$. Then

$$\begin{aligned} & \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_g},\mathsf{n_m}\},o)],\right. \\ & \left.P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_g},\mathsf{n_m}\},o)]\right) \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2}\phi\left(\sum_{\mathsf{n_x}\in\mathcal{N}\setminus N} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_x}\in\mathcal{N}\setminus N} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
&= \frac{1}{2}\phi\left(\sum_{\mathsf{n_x}\in\mathcal{N}} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_x}\in N} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_x}\in\mathcal{N}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_x}\in N} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
&\le \frac{1}{2}\phi\left(\sum_{\mathsf{n_x}\in\mathcal{N}} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_x}\in\mathcal{N}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) + \frac{1}{2}\phi\left(\sum_{\mathsf{n_x}\in N} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_x}\in N} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
&\le \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}\},o)]\right)\\
&+ \frac{1}{2}\sum_{\mathsf{n_x}\in N}\phi\left(P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)
\end{aligned}$$

Thus, overall we get

$$\begin{aligned}
&\sum_{o\in\mathcal{O}_{g,\mathsf{REL}}}\phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)\\
\le &\sum_{\substack{o\in\mathcal{O}_{g,\mathsf{REL}}\\(n_1,n_2,n_3,n_4,n_5):=o}}\left(\sum_{\mathsf{n_g}\in N}\phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}\},o)]\right)\right.\\
&\left.+\frac{1}{2}\sum_{\mathsf{n_x}\in N}\phi\left(P^{0,1}_{n_2,n_3,\mathsf{n_x}}, P^{0,0}_{n_2,n_3,\mathsf{n_x}}\right) + \frac{1}{2}\sum_{\mathsf{n_x}\in N}\phi\left(P^{1,0}_{n_2,n_3,\mathsf{n_x}}, P^{1,1}_{n_2,n_3,\mathsf{n_x}}\right)\right)\\
\le &\sum_{\mathsf{n_g}\in N}\sum_{o\in\mathcal{O}_{g,\mathsf{REL}}}\phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}\},o)]\right)\\
&+\frac{1}{2}\sum_{\substack{\mathsf{n_g}\in N\\\mathsf{n_m}\in N\\\mathsf{n_x}\in N}}\phi\left(P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) + \frac{1}{2}\sum_{\substack{\mathsf{n_g}\in N\\\mathsf{n_m}\in N\\\mathsf{n_x}\in N}}\phi\left(P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)\\
= &\sum_{\mathsf{n_g}\in N}\sum_{o\in\mathcal{O}_{g,\mathsf{REL}}}\phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_g}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_g}\},o)]\right)\\
&+\frac{1}{2}\sum_{\substack{\mathsf{n_g}\in N\\\mathsf{n_x}\in N}}\mathtt{impact}^{(01)(00)}_{\mathtt{indirect}}(\mathsf{n_g},\mathsf{n_x}) + \frac{1}{2}\sum_{\substack{\mathsf{n_g}\in N\\\mathsf{n_x}\in N}}\mathtt{impact}^{(10)(11)}_{\mathtt{indirect}}(\mathsf{n_g},\mathsf{n_x}).
\end{aligned}$$

- $\mathcal{O}_{x,\mathsf{REL}}$. Each observation in this set is of the form $o = (\bot,\bot,\mathsf{n_m},\mathsf{n_x},\mathsf{R}_b)$ and we have $\mathtt{core}(o) = \{\{\mathsf{n_x}\}\}$. We assume $Core = \{\mathsf{n_x}\} \subseteq N$, as otherwise

$$\phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) = \phi(0,0) = 0.$$

Since $\{\mathsf{n_x}\} \in \mathtt{core}(o)$, by Lemmas 4.4.3 and 4.4.4 we get

$$\begin{aligned}
&\phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)\\
\le\ &\phi\left(P^{00,11}[\mathcal{OC}(N\setminus\{\mathsf{n_m}\},o)], P^{01,10}[\mathcal{OC}(N\setminus\{\mathsf{n_m}\},o)]\right)
\end{aligned}$$

$$\begin{aligned}
&= \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_x}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_x},\mathsf{n_m}\},o)],\right. \\
&\quad \left. P^{01,10}[\mathcal{OC}(\{\mathsf{n_x}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_x},\mathsf{n_m}\},o)]\right)
\end{aligned}$$

We now distinguish two cases depending on the recipient R_b:

- Case $\mathsf{R}_b = \mathsf{R}_0$. Then

$$\begin{aligned}
&\phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_x}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_x},\mathsf{n_m}\},o)],\right. \\
&\left. P^{01,10}[\mathcal{OC}(\{\mathsf{n_x}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_x},\mathsf{n_m}\},o)]\right) \\
= &\ \frac{1}{2}\phi\left(\sum_{\mathsf{n_g}\in\mathcal{N}\setminus N} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g}\in\mathcal{N}\setminus N} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \\
= &\ \frac{1}{2}\phi\left(\sum_{\mathsf{n_g}\in\mathcal{N}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_g}\in N} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g}\in\mathcal{N}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_g}\in N} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \\
\leq &\ \frac{1}{2}\phi\left(\sum_{\mathsf{n_g}\in\mathcal{N}} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g}\in\mathcal{N}} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) + \frac{1}{2}\phi\left(\sum_{\mathsf{n_g}\in N} P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g}\in N} P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \\
\leq &\ \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_x}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_x}\},o)]\right) \\
&+\frac{1}{2}\sum_{\mathsf{n_g}\in N}\phi\left(P^{1,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, P^{0,0}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)
\end{aligned}$$

- Case $\mathsf{R}_b = \mathsf{R}_1$. Then

$$\begin{aligned}
&\phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_x}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_x},\mathsf{n_m}\},o)],\right. \\
&\left. P^{01,10}[\mathcal{OC}(\{\mathsf{n_x}\},o) \cap \mathcal{OC}_{\texttt{blank}}(N \setminus \{\mathsf{n_x},\mathsf{n_m}\},o)]\right) \\
= &\ \frac{1}{2}\phi\left(\sum_{\mathsf{n_g}\in\mathcal{N}\setminus N} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g}\in\mathcal{N}\setminus N} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \\
= &\ \frac{1}{2}\phi\left(\sum_{\mathsf{n_g}\in\mathcal{N}} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_g}\in N} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g}\in\mathcal{N}} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}} - \sum_{\mathsf{n_g}\in N} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \\
\leq &\ \frac{1}{2}\phi\left(\sum_{\mathsf{n_g}\in\mathcal{N}} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g}\in\mathcal{N}} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) + \frac{1}{2}\phi\left(\sum_{\mathsf{n_g}\in N} P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, \sum_{\mathsf{n_g}\in N} P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right) \\
\leq &\ \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_x}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_x}\},o)]\right) \\
&+\frac{1}{2}\sum_{\mathsf{n_g}\in N}\phi\left(P^{0,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}, P^{1,1}_{\mathsf{n_g},\mathsf{n_m},\mathsf{n_x}}\right)
\end{aligned}$$

Thus, overall we get (analogously to the case for $\mathcal{O}_{g,\mathsf{REL}}$)

$$\sum_{o\in\mathcal{O}_{x,\mathsf{REL}}} \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)$$

$$\leq \sum_{\mathsf{n_x}\in N}\sum_{o\in\mathcal{O}_{x,\mathsf{REL}}} \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n_x}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n_x}\},o)]\right)$$
$$+\frac{1}{2}\sum_{\substack{\mathsf{n_g}\in N\\ \mathsf{n_x}\in N}} \mathtt{impact}^{(10)(00)}_{\mathtt{indirect}}(\mathsf{n_g},\mathsf{n_x}) + \frac{1}{2}\sum_{\substack{\mathsf{n_g}\in N\\ \mathsf{n_x}\in N}} \mathtt{impact}^{(01)(11)}_{\mathtt{indirect}}(\mathsf{n_g},\mathsf{n_x}).$$

- $\mathcal{O}_{r,\mathsf{REL}}$. This set only contains the observation $o_r = (\bot,\bot,\bot,\bot,\bot)$ and by assumption $P^{00,11}[\mathcal{OC}(N,(\bot,\bot,\bot,\bot,\bot))] \leq P^{01,10}[\mathcal{OC}(N,(\bot,\bot,\bot,\bot,\bot))]$. Thus, $\phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right) = 0$.

As these sets define a partitioning of $\mathcal{O}$, we get

$$\sum_{o\in\mathcal{O}} \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)$$
$$= \sum_{o\in\mathcal{O}_{gx,\mathsf{REL}}} \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)$$
$$+ \sum_{o\in\mathcal{O}_{gm,\mathsf{REL}}} \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)$$
$$+ \sum_{o\in\mathcal{O}_{mx,\mathsf{REL}}} \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)$$
$$+ \sum_{o\in\mathcal{O}_{m}} \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)$$
$$+ \sum_{o\in\mathcal{O}_{g}} \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)$$
$$+ \sum_{o\in\mathcal{O}_{x}} \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)$$
$$+ \sum_{o\in\mathcal{O}_{r}} \phi\left(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\right)$$
$$\leq \sum_{\substack{\mathsf{n}\in N\\ \mathsf{n}'\in N}} \left(\sum_{o\in\mathcal{O}_{gx,\mathsf{REL}}\cup\mathcal{O}_{gm,\mathsf{REL}}\cup\mathcal{O}_{mx,\mathsf{REL}}} \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n},\mathsf{n}'\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n},\mathsf{n}'\},o)]\right)\right)$$
$$+\sum_{\mathsf{n}\in N}\left(\sum_{o\in\mathcal{O}_m} \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n}\},o)]\right)\right.$$
$$+\sum_{o\in\mathcal{O}_g} \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n}\},o)]\right)$$
$$+\frac{1}{2}\sum_{\mathsf{n_x}\in N} \mathtt{impact}^{(01),(00)}_{\mathtt{indirect}}(\mathsf{n},\mathsf{n_x}) + \frac{1}{2}\sum_{\mathsf{n_x}\in N} \mathtt{impact}^{(10),(11)}_{\mathtt{indirect}}(\mathsf{n},\mathsf{n_x})$$
$$+\sum_{o\in\mathcal{O}_x} \phi\left(P^{00,11}[\mathcal{OC}(\{\mathsf{n}\},o)], P^{01,10}[\mathcal{OC}(\{\mathsf{n}\},o)]\right)$$
$$\left.+\frac{1}{2}\sum_{\mathsf{n_g}\in N} \mathtt{impact}^{(10),(00)}_{\mathtt{indirect}}(\mathsf{n_g},\mathsf{n}) + \frac{1}{2}\sum_{\mathsf{n_g}\in N} \mathtt{impact}^{(01),(11)}_{\mathtt{indirect}}(\mathsf{n_g},\mathsf{n})\right)$$

$$\leq \sum_{\mathsf{n}\in N}\Bigg(\mathtt{impact}^{\mathcal{O}}_{\mathsf{REL}}(\{\mathsf{n}\}) + \sum_{\mathsf{n}'\in N}\left(\varepsilon\text{-}\mathtt{impact}^{\mathcal{O}\text{-}2}_{\mathsf{REL}}(\{\mathsf{n},\mathsf{n}'\})\right)$$
$$+\sum_{\mathsf{n}'\in N}\Bigg(+\frac{1}{2}\mathtt{impact}^{(01),(00)}_{\mathtt{indirect}}\left(\mathsf{n},\mathsf{n}'\right) + \frac{1}{2}\mathtt{impact}^{(10),(11)}_{\mathtt{indirect}}\left(\mathsf{n},\mathsf{n}'\right)$$
$$+\frac{1}{2}\mathtt{impact}^{(10),(00)}_{\mathtt{indirect}}\left(\mathsf{n}',\mathsf{n}\right) + \frac{1}{2}\mathtt{impact}^{(01),(11)}_{\mathtt{indirect}}\left(\mathsf{n}',\mathsf{n}\right)\Bigg)\Bigg)$$

□

4.4.6 Approximating the Set of Compromised Nodes

It remains to be shown that the guarantees we compute can be approximated for every budget adversary. To this end we show that the guarantee we get by maximizing the (slightly over-approximated) impact of every node leads to a bound on the advantage of every adversary from the respective class.

Lemma 4.4.8 (Approximating N). *For every anonymity notion α_X with $X \in \{\mathsf{SA},\mathsf{RA},\mathsf{REL}\}$, all senders $\mathsf{S}_0,\mathsf{S}_1 \in \mathcal{S}$, all recipients $\mathsf{R}_0,\mathsf{R}_1 \in \mathcal{R}$, every path selection algorithm* PS*, and every budget B and every cost function f, let A^B_f be any budget adversary that compromises the nodes in the set $N \overset{B,f}{\subseteq} \mathcal{N}$. Then the impact of compromising N can be bounded as follows:*

(1) $\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(N\cup\{\mathsf{R}_0\}) \leq \mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{R}_0)$
$$+ \max_{N^\dagger \overset{B,f}{\subseteq}\mathcal{N}} \sum_{\mathsf{n}\in N^\dagger}\left(\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{n}) + \mathtt{impact}^{\mathtt{ind}}_{\mathsf{SA}}(\mathsf{n},\mathrm{A}^B_f)\right).$$

(2) $\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(N\cup\{\mathsf{S}_0\}) \leq \mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{S}_0)$
$$+ \max_{N^\dagger \overset{B,f}{\subseteq}\mathcal{N}} \sum_{\mathsf{n}\in N^\dagger}\left(\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n}) + \mathtt{impact}^{\mathtt{ind}}_{\mathsf{RA}}(\mathsf{n},\mathrm{A}^B_f)\right).$$

(3) $\mathtt{impact}^{\mathcal{O}}_{\mathsf{REL}}(N) \leq \max_{N^\dagger \overset{B,f}{\subseteq}\mathcal{N}} \sum_{\mathsf{n}\in N^\dagger}\left(\mathtt{impact}^{\mathtt{combined}}_{\mathsf{REL}}(\mathsf{n}) + \mathtt{impact}^{\mathtt{ind}}_{\mathsf{REL}}(\mathsf{n},\mathrm{A}^B_f)\right)$

Proof. Let $\mathsf{S}_0,\mathsf{S}_1 \in \mathcal{S}$ be two senders and $\mathsf{R}_0,\mathsf{R}_1 \in \mathcal{R}$ be two recipients, and let $N' \overset{B,f}{\subseteq} \mathcal{N}$ be any set of compromised Tor nodes within the budget. We show each part separately. In each part we use the fact that $\mathtt{impact}^{(ab),(cd)}_{\mathtt{indirect}}(\mathsf{n},\mathsf{n}) = 0$, which holds since a node cannot occur in the same circuit twice with non-zero probability.

(1) By Lemma 4.4.5 we know that for $N = N'\cup\{\mathsf{R}_0\}$ we have

$$\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(N)$$

$$\begin{aligned}
&= \sum_{o\in\mathcal{O}} \phi\big(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\big)\\
&\leq \mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{R}_0) + \sum_{\mathsf{n}\in N'} \Big(\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{n}) + \mathtt{impact}_{\mathtt{Rec1}}(\mathsf{n})\\
&\quad + \sum_{\mathsf{n}'\in N'\setminus\{\mathsf{n}\}} \Big(\mathtt{impact}^{(10)(00)}_{\mathtt{indirect}}(\mathsf{n},\mathsf{n}') + \mathtt{impact}_{\mathtt{Rec2(n,n')}}\Big)\Big)\\
&\leq \mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{R}_0) + \max_{N^\dagger \overset{B,f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}\in N^\dagger} \Big(\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{n}) + \mathtt{impact}_{\mathtt{Rec1}}(\mathsf{n})\\
&\quad + \max_{N^* \overset{B-f(\mathsf{n}),f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}'\in N^*} \Big(\mathtt{impact}^{(10)(00)}_{\mathtt{indirect}}(\mathsf{n},\mathsf{n}') + \mathtt{impact}_{\mathtt{Rec2(n,n')}}\Big)\Big)\\
&= \mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{R}_0) + \max_{N^\dagger \overset{B,f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}\in N^\dagger} \big(\mathtt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{n}) + \mathtt{impact}^{\mathtt{ind}}_{\mathsf{SA}}(\mathsf{n}, \mathrm{A}^B_f)\big)\,.
\end{aligned}$$

(2) By Lemma 4.4.6 we know that for $N = N' \cup \{\mathsf{S}_0\}$ we have

$$\begin{aligned}
&\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(N)\\
&= \sum_{o\in\mathcal{O}} \phi\big(P^{00}[\mathcal{OC}(N,o)], P^{10}[\mathcal{OC}(N,o)]\big)\\
&\leq \mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{S}_0) + \sum_{\mathsf{n}\in N'} \Big(\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n}) + \mathtt{impact}_{\mathtt{Sen1}}(\mathsf{n})\\
&\quad + \sum_{\mathsf{n}'\in N'\setminus\{\mathsf{n}\}} \Big(\mathtt{impact}^{(01),(00)}_{\mathtt{indirect}}(\mathsf{n},\mathsf{n}') + \mathtt{impact}_{\mathtt{Sen2}}(\mathsf{n},\mathsf{n}')\Big)\Big)\\
&\leq \mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{S}_0) + \max_{N^\dagger \overset{B,f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}\in N^\dagger} \Big(\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n}) + \mathtt{impact}_{\mathtt{Sen1}}(\mathsf{n})\\
&\quad + \max_{N^* \overset{B-f(\mathsf{n}),f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}'\in N^*} \Big(\mathtt{impact}^{(01),(00)}_{\mathtt{indirect}}(\mathsf{n}',\mathsf{n}) + \mathtt{impact}_{\mathtt{Sen2}}(\mathsf{n},\mathsf{n}')\Big)\Big)\\
&= \mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{S}_0) + \max_{N^\dagger \overset{B,f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}\in N^\dagger} \big(\mathtt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n}) + \mathtt{impact}^{\mathtt{ind}}_{\mathsf{RA}}(\mathsf{n}, \mathrm{A}^B_f)\big)\,.
\end{aligned}$$

(3) By Lemma 4.4.7 we know that for $N = N'$ we have

$$\begin{aligned}
&\sum_{o\in\mathcal{O}} \phi\big(P^{00,11}[\mathcal{OC}(N,o)], P^{01,10}[\mathcal{OC}(N,o)]\big)\\
\leq &\sum_{\mathsf{n}\in N} \Bigg(\mathtt{impact}^{\mathcal{O}}_{\mathsf{REL}}(\{\mathsf{n}\}) + \sum_{\mathsf{n}'\in N} \big(\varepsilon\text{-}\mathtt{impact}^{\mathcal{O}\text{-}2}_{\mathsf{REL}}(\{\mathsf{n},\mathsf{n}'\})\big)
\end{aligned}$$

$$
\begin{aligned}
& + \sum_{\mathsf{n}' \in N} \Bigg(+ \frac{1}{2} \mathtt{impact}^{(01),(00)}_{\mathtt{indirect}} \left(\mathsf{n}, \mathsf{n}'\right) + \frac{1}{2} \mathtt{impact}^{(10),(11)}_{\mathtt{indirect}} \left(\mathsf{n}, \mathsf{n}'\right) \\
& + \frac{1}{2} \mathtt{impact}^{(10),(00)}_{\mathtt{indirect}} \left(\mathsf{n}', \mathsf{n}\right) + \frac{1}{2} \mathtt{impact}^{(01),(11)}_{\mathtt{indirect}} \left(\mathsf{n}', \mathsf{n}\right) \Bigg) \Bigg) \\
= & \sum_{\mathsf{n} \in N} \Bigg(\mathtt{impact}^{\mathcal{O}}_{\mathsf{REL}}(\{\mathsf{n}\}) + \sum_{\mathsf{n}' \in N \setminus \mathsf{n}} \left(\varepsilon\text{-}\mathtt{impact}^{\mathcal{O}\text{-}2}_{\mathsf{REL}}(\{\mathsf{n}, \mathsf{n}'\}) \right) \\
& + \sum_{\mathsf{n}' \in N \setminus \mathsf{n}} \Bigg(+ \frac{1}{2} \mathtt{impact}^{(01),(00)}_{\mathtt{indirect}} \left(\mathsf{n}, \mathsf{n}'\right) + \frac{1}{2} \mathtt{impact}^{(10),(11)}_{\mathtt{indirect}} \left(\mathsf{n}, \mathsf{n}'\right) \\
& + \frac{1}{2} \mathtt{impact}^{(10),(00)}_{\mathtt{indirect}} \left(\mathsf{n}', \mathsf{n}\right) + \frac{1}{2} \mathtt{impact}^{(01),(11)}_{\mathtt{indirect}} \left(\mathsf{n}', \mathsf{n}\right) \Bigg) \Bigg) \\
\leq & \max_{N^\dagger \overset{B,f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n} \in N^\dagger} \Bigg(\mathtt{impact}^{\mathcal{O}}_{\mathsf{REL}}(\{\mathsf{n}\}) + \max_{N^* \overset{B-f(\mathsf{n}),f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}' \in N^*} \left(\varepsilon\text{-}\mathtt{impact}^{\mathcal{O}\text{-}2}_{\mathsf{REL}}(\{\mathsf{n}, \mathsf{n}'\}) \right) \\
& + \max_{N^* \overset{B-f(\mathsf{n}),f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n}' \in N^*} \Bigg(+ \frac{1}{2} \mathtt{impact}^{(01),(00)}_{\mathtt{indirect}} \left(\mathsf{n}, \mathsf{n}'\right) + \frac{1}{2} \mathtt{impact}^{(10),(11)}_{\mathtt{indirect}} \left(\mathsf{n}, \mathsf{n}'\right) \\
& + \frac{1}{2} \mathtt{impact}^{(10),(00)}_{\mathtt{indirect}} \left(\mathsf{n}', \mathsf{n}\right) + \frac{1}{2} \mathtt{impact}^{(01),(11)}_{\mathtt{indirect}} \left(\mathsf{n}', \mathsf{n}\right) \Bigg) \Bigg) \\
= & \max_{N^\dagger \overset{B,f}{\subseteq} \mathcal{N}} \sum_{\mathsf{n} \in N^\dagger} \left(\mathtt{impact}^{\mathtt{combined}}_{\mathsf{REL}}(\mathsf{n}, \mathrm{A}^B_f) + \mathtt{impact}^{\mathtt{ind}}_{\mathsf{REL}}(\mathsf{n}, \mathrm{A}^B_f) \right)
\end{aligned}
$$

□

Lemma 4.4.8 concludes our proof. We have shown that our anonymity guarantees indeed are sound upper bounds on the anonymity impact of a budget adversary.

Chapter 5

Evaluation

In this final chapter we show the applicability of our methodology, by calculating formal guarantees for Tor, depending on its path selection algorithm PS against a variety of AnoA adversaries. Based on the foundation of Chapter 2 and the calculations of Chapter 4 we perform two analyses. First, we analyze Tor against a multitude of *node budget adversaries*, i.e., adversaries that may compromise a set of Tor nodes, based on their parameters, including k-collusion adversaries, bandwidth restricted adversaries, economical adversaries and country restricted adversaries.

Second, we construct a model of Tor's Internet topology and evaluate Tor against several *network infrastructure adversaries*, including malicious companies that provide autonomous systems (AS), malicious Internet exchange points (IXP) and malicious submarine cables.

5.1 Evaluating Tor Against Node Adversaries

One of the greatest concerns of Tor users is that some of the large number of Tor nodes are malicious. In this part of our evaluation we analyze the impact of such eavesdropping Tor nodes on anonymity. To this end, we first instantiate the budget adversaries from Section 4.3. We then describe the setup of the evaluation (i.e., the senders and recipients we considered and the gathering of Tor consensus information), present a few technical details about the implementation of our analysis tool and finally show and discuss the results of our large-scale analysis.

5.1.1 Instantiating Node Budget Adversaries

We consider the following four instances of budget adversaries in our analysis and evaluate their anonymity impact on Tor's anonymity for all considered path selection algorithms.

k-collusion adversary. We evaluate the k-collusion adversary for up to 25 compromised nodes, i.e., for a budget B ranging from 0 to 25. Recall that the k-collusion adversary is restricted only by the number of compromised nodes, independently of their properties. Consequently, it will choose very influential Tor nodes.

Bandwidth adversary. We evaluate the bandwidth adversary for a budget B ranging from 1 MB/s to 1 GB/s. Recall that the bandwidth adversary is not restricted in the number of nodes it can compromise. Thus, it may either compromise a few influential nodes with high bandwidth, or many less influential nodes with low bandwidth. Its decision is based mainly on the ratio between impact and bandwidth.

Geographic adversary. We evaluate several adversaries that compromise all nodes in a given country. We consider the five top countries according to offered Tor bandwidth: Germany (DE), France (FR), the Netherlands (NL), the United States (US) and the United Kingdom (GB). Recall that as a predicate adversary, the geographic adversary can exactly compromise the Tor nodes for which the predicate is true. Since compromising any node can only increase the adversary's advantage, the geographic adversary will compromise all nodes located in the respective country, independently of their number or their other properties.

Monetary adversary. We evaluate a monetary adversary with a monthly budget B in US dollars, ranging from 10^3 to 10^8 US dollars. Recall that the cost function $f_\$$ assigns each node its monthly cost, depending on a price function $\texttt{price}(\texttt{n.provider}, \texttt{n.BW})$. We instantiate `price` for the 8 largest providers hosting Tor nodes (Amazon, DigitalOcean, Hetzner, LeaseWeb, myLoc, Online, OVH, and STRATO), accounting for approximately $\frac{1}{3}$ of Tor bandwidth. We assign each node hosted by one of these providers the price of the cheapest product of the respective provider that offers at least the node's average bandwidth. For all remaining nodes (that are not hosted by these providers), $\texttt{price}(\cdot)$ assigns the average consumer prices per bandwidth, depending on the node's country, taken from Ookla's NetIndex [90] per country. The monetary adversary will mainly base its decision on the ratio between the impact of nodes and their costs, i.e., it is mainly concerned with the ratio between the impact and the price that the respective provider asks for a server with the necessary bandwidth.

5.1.2 Setup

Recall that our computations are with respect to specific senders $\mathsf{S}_0, \mathsf{S}_1$ and recipients $\mathsf{R}_0, \mathsf{R}_1$. For the sake of evaluation, we hence consider concrete users in the following: the IP addresses from the affiliations of the PC chairs of PETS2015 and PETS2016. The first user, S_0, establishes a Tor circuit from Drexel University in Philadelphia; the second user, S_1, connects from Indiana University in Bloomington. As possible destinations, we have selected TU Darmstadt as R_0 and KU Leuven as R_1. For both destinations, we only required the HTTPS port 443 as the by far most widely used port for Tor connections (Tor is mainly used via the Tor-Browser bundle, which includes HTTPS-Everywhere). We added one analysis where the recipient instead requires the TCP port 6667 (used for IRC chat) to show the impact of ports on (recipient) anony-

```
MAXIMIZATION($B, f, \mathcal{N}$, [$\texttt{impact}(\mathsf{n})$ for $\mathsf{n} \in \mathcal{N}$])
 1: OUT:=0, $L$:=$\emptyset$
 2: for $\mathsf{n} \in \mathcal{N}$ do
 3:     if $f(\mathsf{n}) = 0$ then
 4:         OUT:=OUT + $\texttt{impact}(\mathsf{n})$
 5:     else if $f(\mathsf{n}) \leq B$ then
 6:         $L$:=$L \cup \{\mathsf{n}\}$
 7:     end if
 8: end for
 9: $B'$:=$B$
10: for $\mathsf{n} \in L$, ordered by $\frac{\texttt{impact}(\mathsf{n})}{f(\mathsf{n})}$ (descending) do
11:     if $B' \geq f(\mathsf{n})$ then
12:         OUT:=OUT + $\texttt{impact}(\mathsf{n})$
13:         $B'$:=$B' - f(\mathsf{n})$
14:     else
15:         OUT:=OUT + $\frac{B'}{f(\mathsf{n})} \cdot \texttt{impact}(\mathsf{n})$
16:         $B'$:=0
17:     end if
18: end for
19: return OUT
```

Figure 5.1: Simple maximization algorithm for *MATor*.

mity. Moreover, since the choice of the IP address mainly influences LASTor, we add a second setup for LASTor (which we coin LASTor (2), or L2 in our graphs), in which we swap the IP addresses of S_1 and R_1.

Unless stated otherwise in a specific experiment, we used the following data in our evaluation: Our evaluation was conducted on Tor network consensus data over the course of almost one year (August 2014-May 2015). For each month we gathered 15 consensus files (resulting in around one file every second day, at noon) and calculated the anonymity impact of the considered adversaries.

5.1.3 Implementation

We have implemented the computation of $\texttt{impact}_X$ as a C++ tool, which we coin *MATor*. *MATor* takes care of computing the probability distribution over Tor circuits for a given Tor network consensus and the respective server descriptors. Moreover, we added the calculations from Chapter 4 for any given budget adversary $\mathcal{A}_f^B(\cdot)$, the desired anonymity notion α_X, and concrete senders $\mathsf{S}_0, \mathsf{S}_1$ and recipients $\mathsf{R}_0, \mathsf{R}_1$. To this end, we first compute the individual impacts $\texttt{impact}^{\mathcal{O}}$ for all observations of individual nodes, pairs of nodes and – depending on the anonymity notion – relevant end points. Leveraging the computation from $\texttt{impact}_X^{\mathcal{O}}$ to $\texttt{impact}_X$ requires us to solve the underlying integer maximization problems, e.g., determining $N \subseteq \mathcal{N}$ such that $\sum_{n\in N} f(n) \leq B$ becomes maximal. While this problem is known to be NP-hard, we soundly approximate it via a simple optimization algorithm. Our algorithm chooses the nodes with the best ratio of impact and costs to consume up the budget. As soon as a node is too expensive, the algorithm adds a fraction of the node's bandwidth to the overall impact, depending on the remaining budget and the costs. As we order the nodes by their impact, this method soundly converts the budget into an approximated overall impact. We refer to Fig. 5.1 for a description of this algorithm. The source code of our implementation is publicly available [103].

About Proofs, Implementations and Good Scientific Practice While working on the write-up of this thesis we realized that our previous formalizations did not correctly capture the (slightly unintuitive) indirect impact of a node. Moreover, the implementation of our *MATor* tool had several minor bugs and also did not correctly capture this impact. Consequently, we refactored our *MATor* implementation and, along the way, also added the possibility for calculating lower bounds to measure the margin of error of our calculations. Subsequently we repeated the evaluation that already appeared in [12] for this thesis. To compare our results with the evaluation presented there, we decided to re-evaluate the time period chosen in our paper.

5.1.4 Results

To allow for parsing the large number of results, we try to present them in a consistent notation and divide the results into intuitive categories. First we present the impact of our adversaries on Tor's current path selection algorithm. The respective graphs present a ground truth for our comparison with other path selection algorithms. To simplify understanding the impact on other path selection algorithms, we then provide both the anonymity impact of the adversaries on the alternative path selection algorithms and difference-graphs, i.e., graphs that show how the anonymity guarantee for the respective path selection algorithm compares to Tor's path selection algorithm.

Moreover, we present the impact of alternative path selection algorithms in the *transition phase*, in which some few users use the alternative while most users stick to Tor's current path selection algorithm.

Evaluating Tor's Path Selection Algorithm

The results of our evaluation of TorPS are depicted in Figures 5.2 and 5.3.

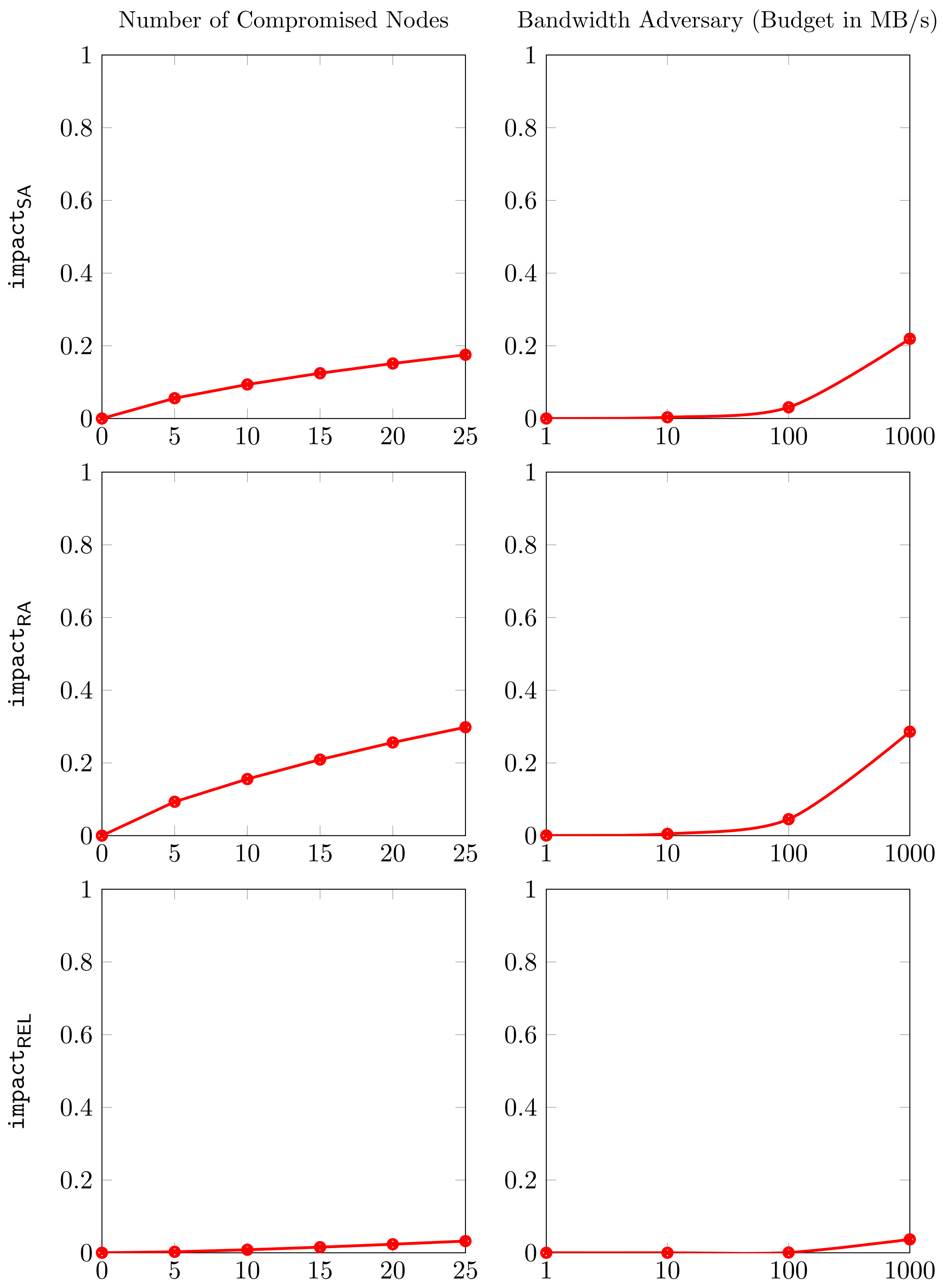


Figure 5.2: Adversarial `impact` for sender (top), recipient (middle) and relationship (bottom) anonymity for the k-collusion adversary for k from 0 to 25 (left) and the bandwidth adversary with a budget from 1 MB/s up to $1,000$ MB/s (right).

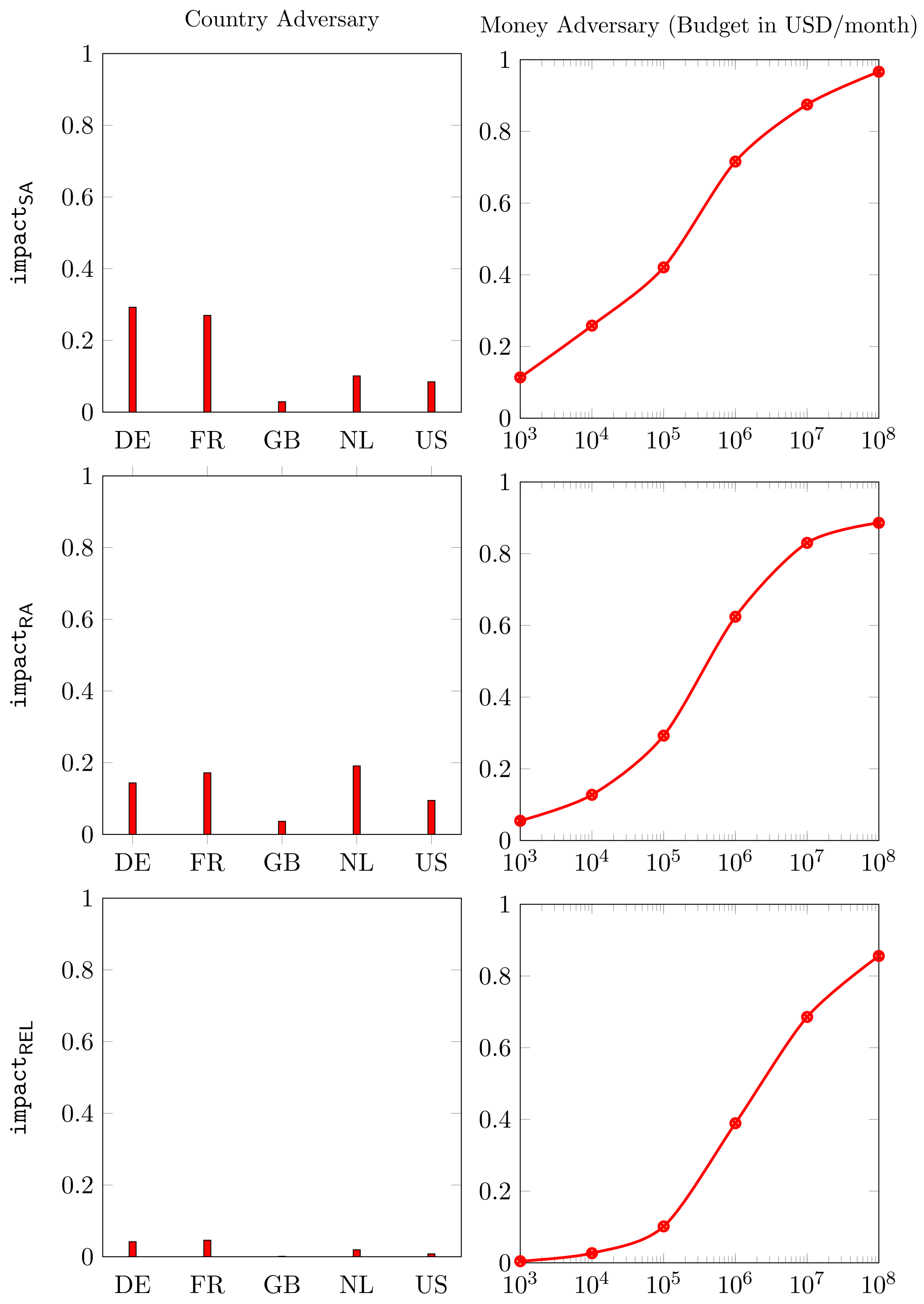


Figure 5.3: Adversarial `impact` for sender (top), recipient (middle) and relationship (bottom) anonymity for the geographic adversary for the countries Germany (DE), France (FR), the United Kingdom (GB), Netherlands (NL) and the United States (US) (left), and the monetary adversary with a monetary budget in USD/month (right).

Results – k-collusion adversary (left, Fig. 5.2) Our results confirm that a small number of influential nodes suffices to significantly reduce anonymity. Allowing the adversary to compromise 20 nodes of its choice leads to a 15.1% reduction of sender anonymity, a 25.6% reduction of recipient anonymity and a 2.3% reduction of relationship anonymity. This reduction of anonymity increases sublinearly for sender anonymity and recipient anonymity, which we attribute to the fact that for Tor's path selection algorithm alone there is no indirect impact and thus the advantage is considered on a per-node basis. Hence, the adversary will choose the node with the highest impact first and every upcoming node can only have a smaller impact than any of the previous nodes. For relationship anonymity however, every additional nodes can increase the impact of any already compromised node, as the pairwise impact is quite significant.

Results – bandwidth adversary (right, Fig. 5.2) The plot shows the reduction of anonymity on a logarithmic scale axis. Our results confirm that a significant amount of bandwidth is necessary for reducing Tor's anonymity. An adversary that may compromise nodes with an overall advertised bandwidth of 1GB/s leads to a 21.9% reduction of sender anonymity, a 28.6% reduction of recipient anonymity and a 3.7% reduction of relationship anonymity. We see that reducing sender anonymity in this scenario is, in comparison to the k-collusion adversary, slightly easier in comparison to reducing recipient anonymity.

Results – geographic adversary (left, Fig. 5.3) Our plot shows the impact of the top five countries according to offered Tor bandwidth: Germany (DE), France (FR), the Netherlands (NL), the United States (US) and the United Kingdom (GB). We confirm that, although the largest number of Tor nodes is within the United States, the geographic adversaries for Germany, France and the Netherlands reduce anonymity more than the geographic adversary for the United States, which we attribute to the fact that although more Tor nodes are within the US, their bandwidth and thus the probability that they are used is significantly smaller. The Germany adversary reduces sender anonymity most drastically (by 29.2%), whereas the Netherlands adversary reduces recipient anonymity most drastically (by 19.1%), and the France adversary reduces relationship anonymity most drastically (by 4.6%). We attribute this to the large guard bandwidth in Germany, the large exit bandwidth of the Netherlands and the combination of a high guard bandwidth and high exit bandwidth in France.

Results – monetary adversary (right, Fig. 5.3) The plot shows the impact of our monetary adversary for a (monthly) budget of 1,000 up to 100 Mio USD. We observe that an adversary with a rather small budget of only 1,000 USD per month reduces sender anonymity by 11.4% and recipient anonymity by 5.5%. An adversary with a monthly budget of 100,000 USD reduces sender anonymity by 42.1%, recipient

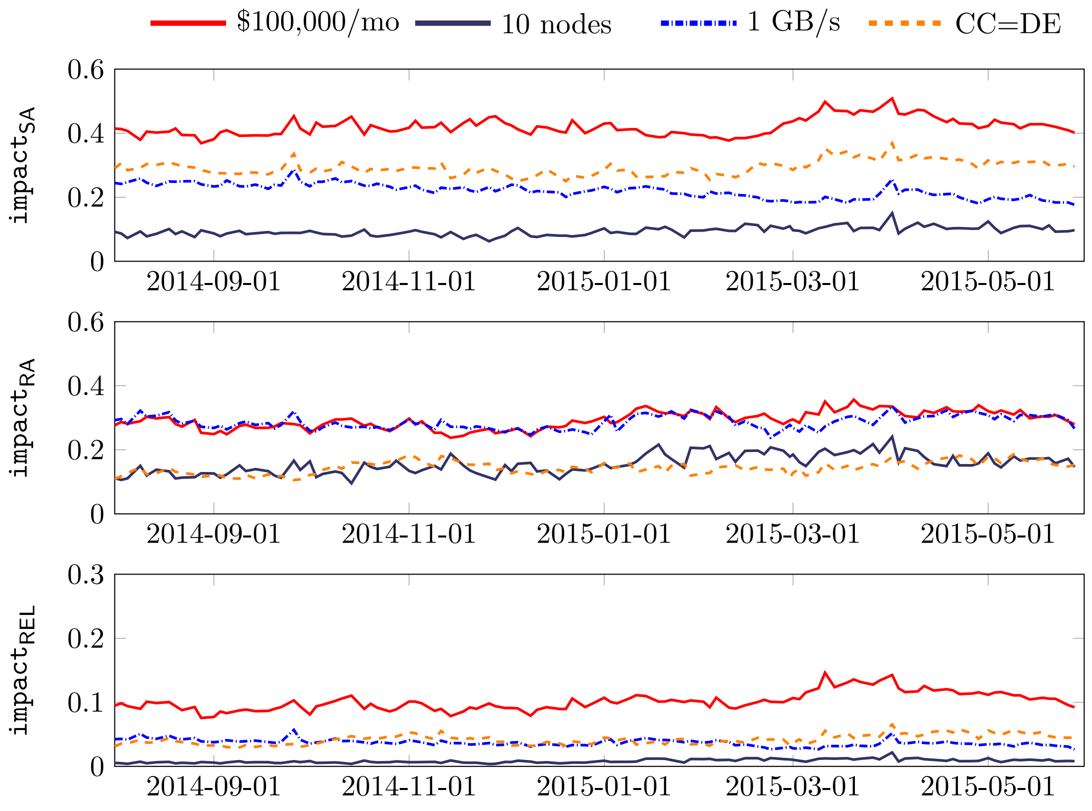


Figure 5.4: Changes in the adversarial `impact` against selected adversarial strategies for Tor's path selection during the last year. Note, that the Y axis scales from 0 to 0.6 for sender and recipient anonymity plots, and from 0 to 0.3 for relationship anonymity.

anonymity by 29.3% and relationship anonymity by 10.2%. We attribute this to a significant number of Tor nodes hosted by OVH and similar companies that offer very cheap bandwidth.

Guarantees Over Time

The anonymity guarantees we can derive depend not only on the specification of the path selection algorithm and the adversary, but also, significantly, on the status of the Tor network. We included a plot that shows the change in anonymity guarantees for a selection of four adversaries and for Tor's current path selection algorithm. The adversaries we included are a k-collusion adversary with $k = 10$, a bandwidth adversary with a budget of 1 GB/s, a geographical adversary that compromises all Tor nodes in Germany and a monetary adversary with a monthly budget of $100,000$ USD.

Our results confirm that the Tor network is subject to a constant fluctuation of nodes, which impacts the anonymity guarantees we calculate. While some of the fluctuations only impact the reduction of anonymity by individual classes of adversaries, there are fluctuations that impact the reduction of anonymity of all considered adversaries at once.

Alternative Path Selection Algorithms

The results of our evaluation of alternative path selection algorithms are depicted in Figures 5.5 to 5.8 – from top to bottom: sender anonymity, recipient anonymity, and relationship anonymity. In Figures 5.5 and 5.6 we show the differences between the anonymity impacts of the alternative path selection algorithms and TorPS and, for completion, in Figures 5.7 and 5.8 we show the results we obtained for the alternative path selection algorithms directly.

UniformTor It is commonly believed that the uniform distribution over all (eligible) nodes offers the highest degree of anonymity. We can confirm that against k-collusion adversaries, this is certainly true. Against k-collusion adversaries the uniform path selection algorithm excels: such an adversary with a budget of $k = 20$ Tor nodes only reduces sender anonymity by 1.3%, recipient anonymity by 1.9% and relationship anonymity by $< 0.1\%$.

However, an adversary that corrupts a certain amount of bandwidth can corrupt a large number of low-bandwidth nodes even with a small budget. Consequently, the bandwidth adversary with a budget of 1 GB/s reduces sender anonymity by 27.1%, recipient anonymity by 56.7% and relationship anonymity by 12.6%, which, especially for recipient anonymity and relationship anonymity is significantly worse in comparison with Tor's path selection algorithm. The geographic adversaries also reduce anonymity differently, which we attribute to the difference in the number of Tor nodes in a country and the bandwidth of these nodes, i.e., the United States adversary reduces anonymity significantly more for the uniform path selection algorithm than for Tor's path selection algorithm. In case of a monetary adversary, the uniform path selection algorithm offers slightly better sender anonymity guarantees, but significantly worse recipient anonymity guarantees. We attribute this to a number of relatively cheap, high-bandwidth guards and a large number of low-bandwidth exit nodes.

SelekTOR As SelekTOR only restricts the choice for an exit node, sender anonymity is comparable to Tor, while relationship anonymity and in particular recipient anonymity suffer significantly, for essentially all considered adversaries, except for the geographic adversaries outside of the United States. As expected, the geographic adversary that compromises all nodes within the US can break recipient anonymity with 100% probability (it always controls the exit node and can perform a traffic correlation attack).

A noteworthy exception to our observations is that for a low budget, the monetary adversary reduces recipient anonymity significantly less, which we attribute to a higher

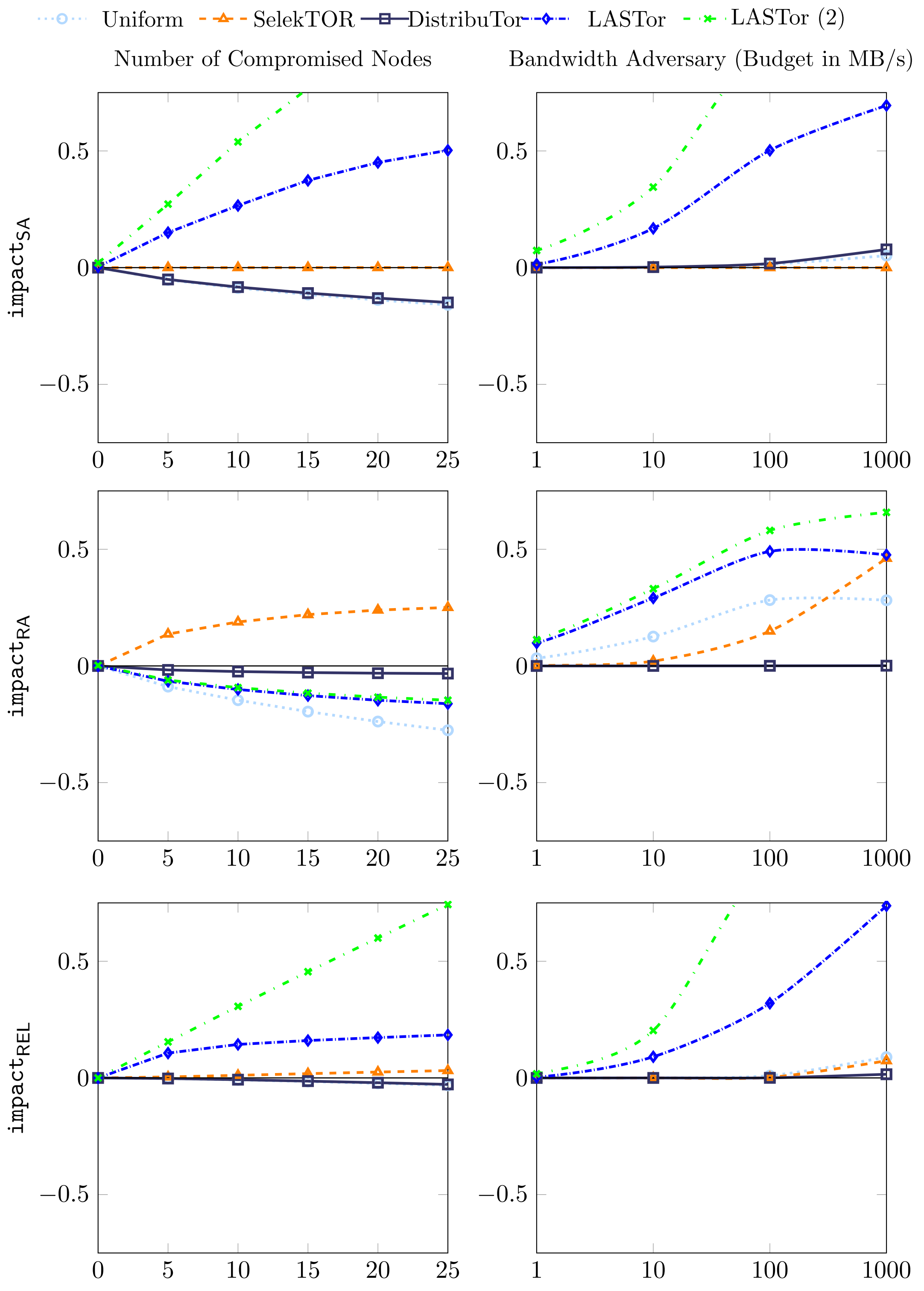


Figure 5.5: The difference between $\texttt{impact}_X$ for sender (top), recipient (middle) and relationship (bottom) anonymity of alternative path selection algorithms and TorPS. The Considered adversaries classes are the k-collusion adversary for k from 0 to 25 (left) and the bandwidth adversary with a budget from 1 MB/s up to 1,000 MB/s (right).

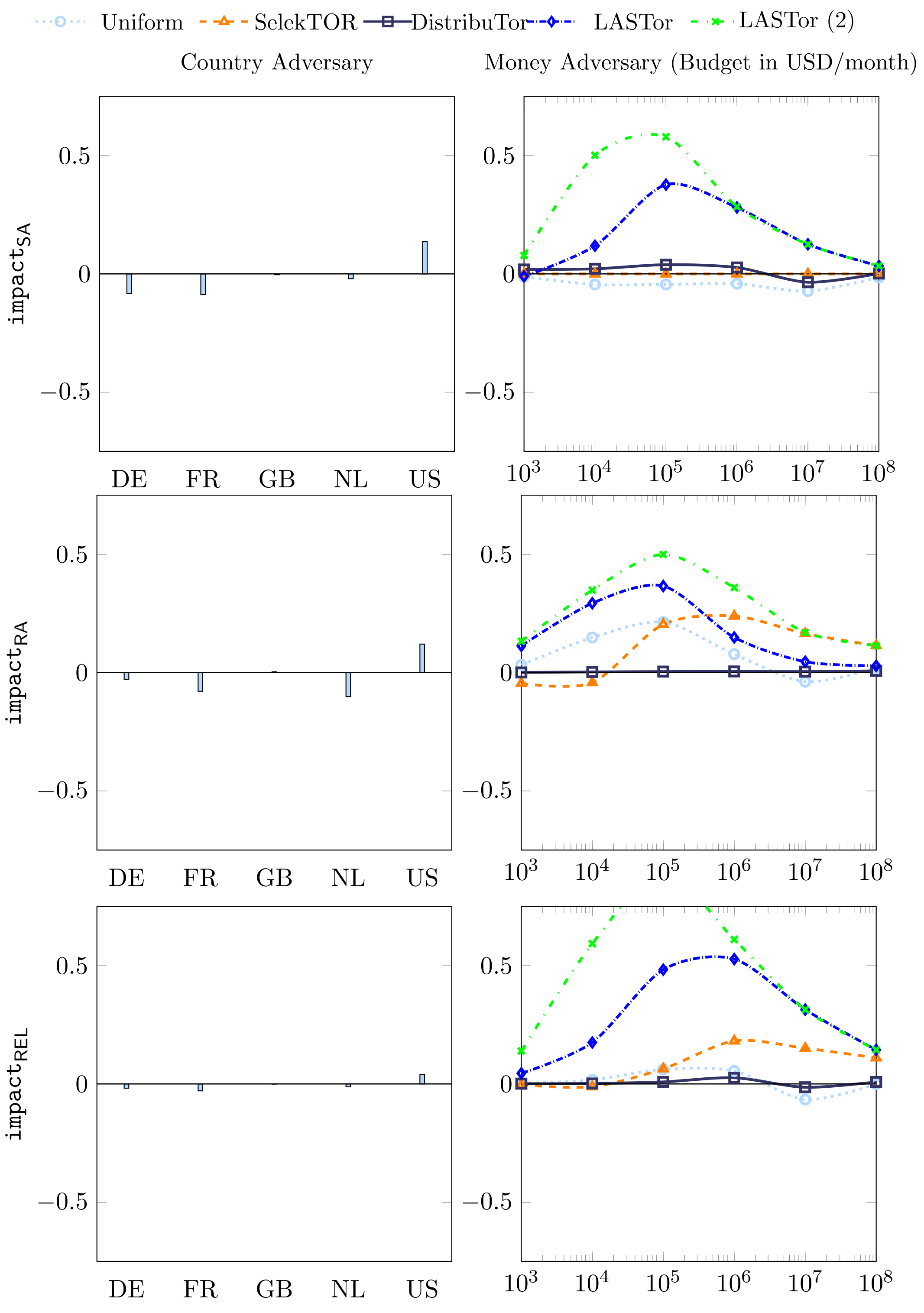


Figure 5.6: The difference between $\texttt{impact}_X$ for sender (top), recipient (middle) and relationship (bottom) anonymity of alternative path selection algorithms and TorPS. The considered adversary classes are the geographic adversary for the countries Germany (DE), France (FR), the United Kingdom (GB), Netherlands (NL) and the United States (US) (left), and the monetary adversary with a monetary budget in USD/month (right).

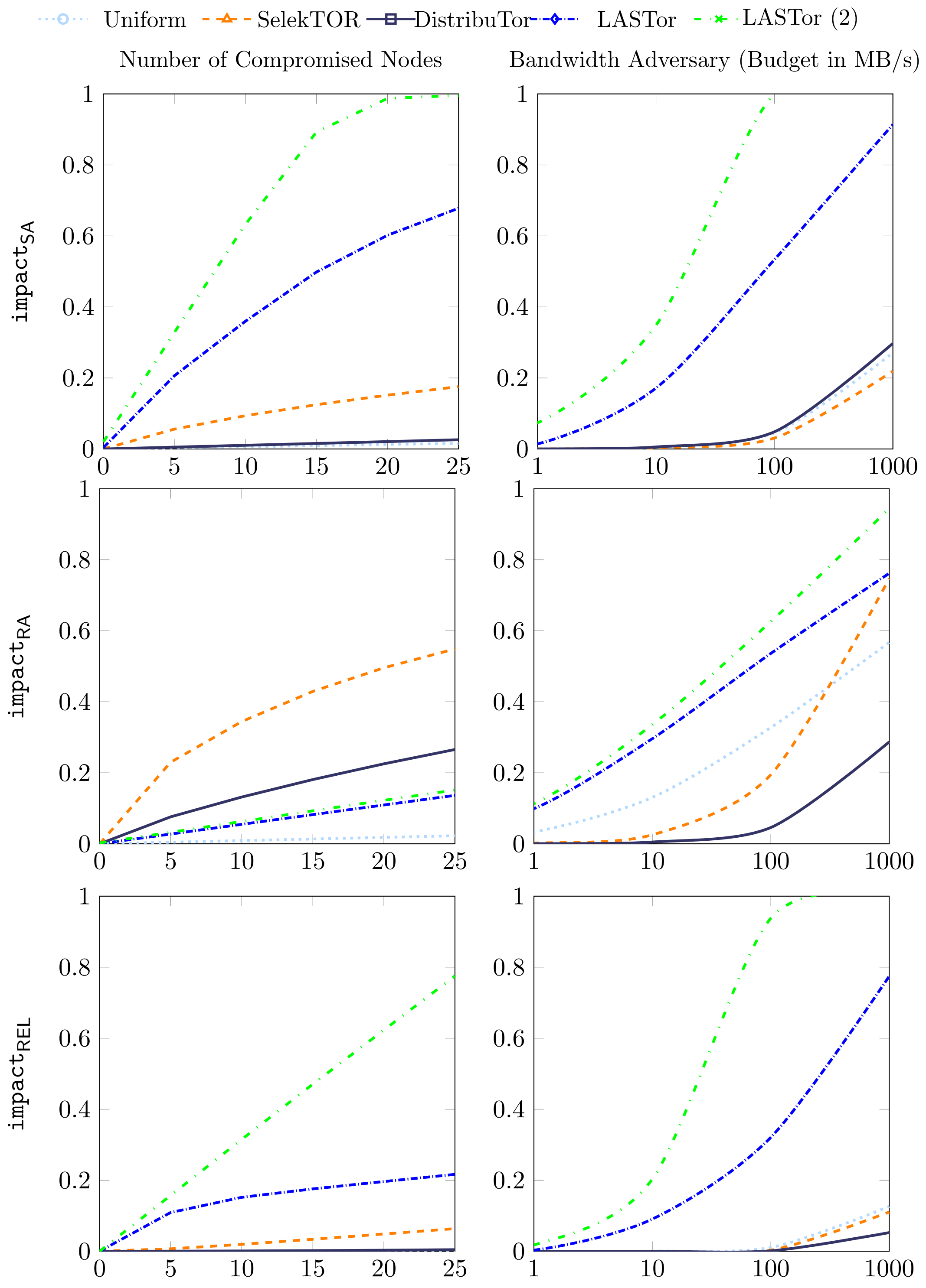


Figure 5.7: Adversarial `impact` for sender (top), recipient (middle) and relationship (bottom) anonymity for all considered alternative path selection algorithms. The Considered adversaries classes are the k-collusion adversary for k from 0 to 25 (left) and the bandwidth adversary with a budget from 1 MB/s up to 1,000 MB/s (right).

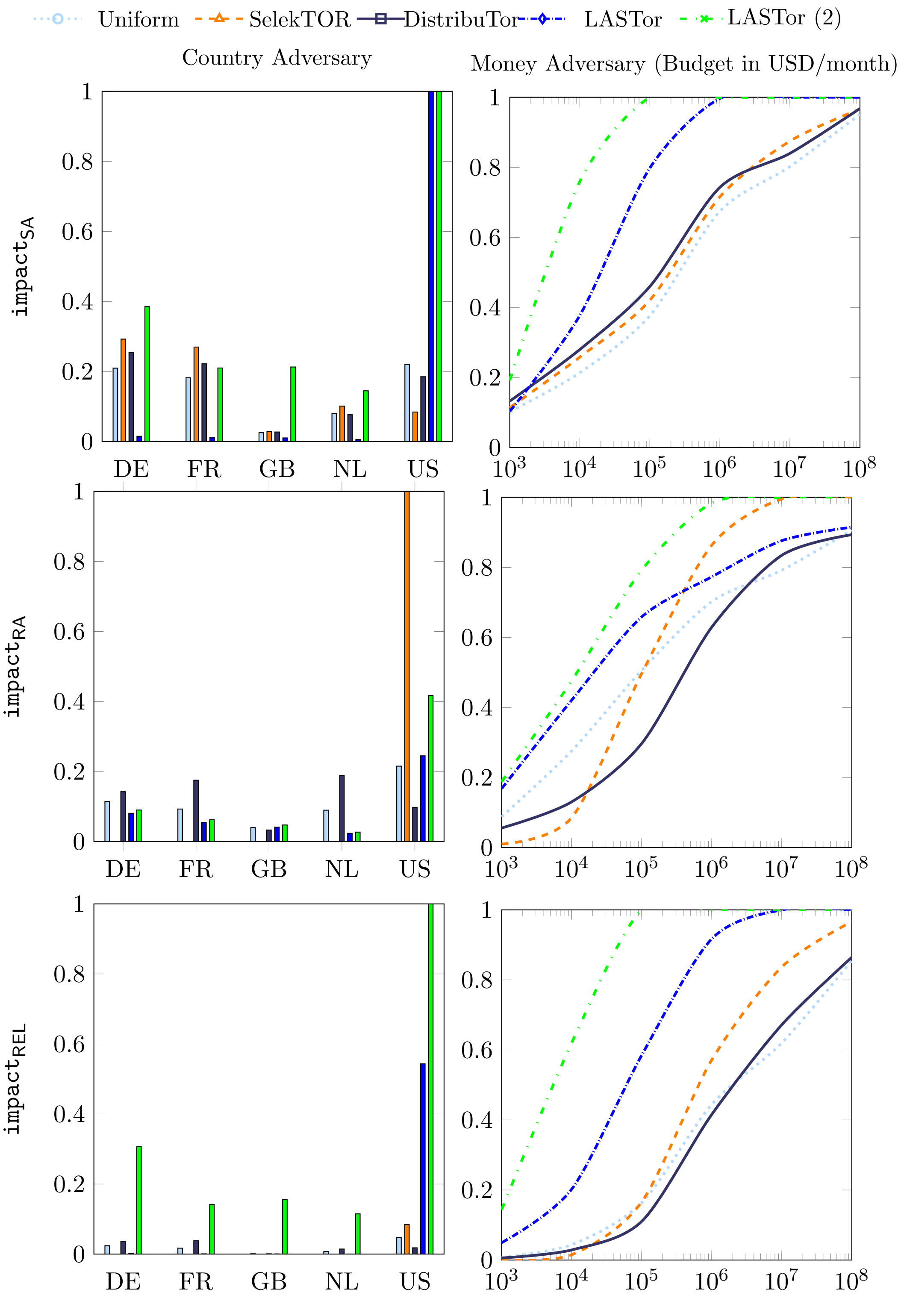


Figure 5.8: Adversarial `impact` for sender (top), recipient (middle) and relationship (bottom) anonymity for all considered alternative path selection algorithms. The considered adversary classes are the geographic adversary for the countries Germany (DE), France (FR), the United Kingdom (GB), Netherlands (NL) and the United States (US) (left), and the monetary adversary with a monetary budget in USD/month (right).

average price for hosting small Tor nodes in the United States in comparison with some cheap European providers.

DistribuTor DistribuTor in particular modifies the selection of entry nodes by capping the possible weights of nodes at a certain point. Consequently, it achieves much better sender anonymity and relationship anonymity guarantees against k-collusion adversaries (against which it has been designed), up to the point of being comparable with the uniform path selection algorithm. However, against bandwidth adversaries it suffers a higher reduction of sender anonymity and relationship anonymity. Since it uses modified weights (especially for entry nodes), its sender anonymity is, in comparison to Tor's sender anonymity, slightly less prone against European country adversaries, but more vulnerable against the US country adversary (there are more smaller entry nodes in the US). Our guarantees for recipient anonymity are very comparable to the guarantees for Tor's current path selection algorithm, which we attribute to the bottleneck of exit bandwidth in Tor: Tor's current path selection algorithm already uses possible exit nodes (almost) solemnly for the exit position, and thus DistribuTor does not significantly change the choice of exit nodes.

LASTor Since our node adversaries do not perform network-based attacks, our evaluation of LASTor, which is designed to counter network-based attacks, may seem slightly unfair. We refer to Section 5.2 for an analysis of LASTor against compromised network infrastructure. The uniform distribution of the node weights within LASTor's "geolocation buckets", leads to significantly better results for recipient anonymity against a k-collusion adversary due to the large number of low-bandwidth exit nodes. The reduction of sender anonymity, however, is much more severe, as the adversary can easily gain partial information. Moreover, even a small amount of compromised bandwidth suffices for completely reducing anonymity, as even a small, compromised middle node can gain information about the location of both sender and recipient of a communication: entry and exit nodes have significantly different weights for different locations. Note that LASTor (as presented in [1]) additionally restricts circuits depending on whether traffic is expected to be routed through the same autonomous system twice. We did not capture this aspect for our analysis in this section, as we wanted to avoid an influence of incomplete information (about the network infrastructure) on our analysis. Capturing this additional restriction, however, would only increase the advantage of an eavesdropping node adversary.

To further evaluate LASTor, we also ran analyses in which we swapped the IP address of one recipient with the IP address of one sender (TU Darmstadt with Indiana University). This is displayed as *LASTor (2)* in Figures 5.5 and 5.6. As expected, the resulting more diverse distribution of senders and recipients has a negative impact on the anonymity guarantees.

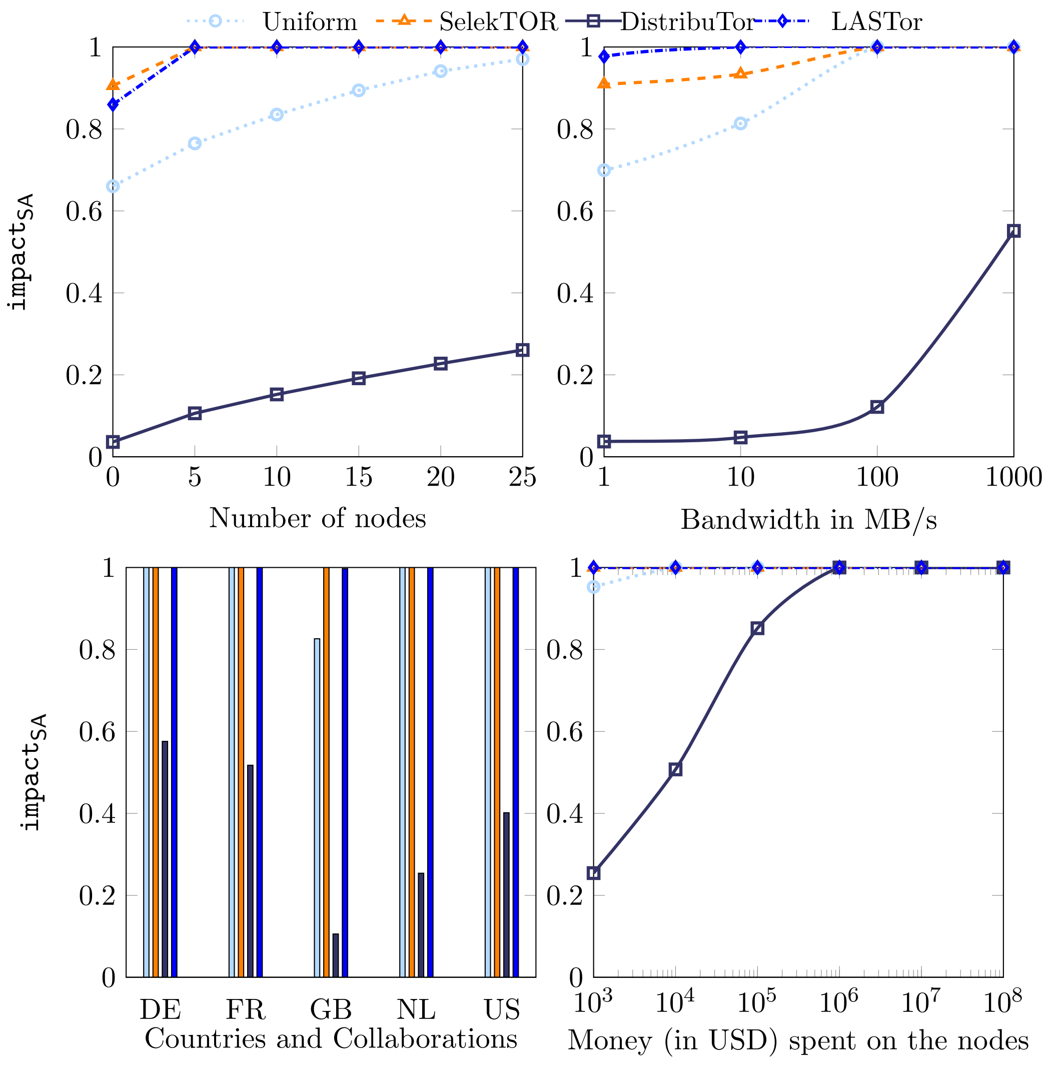


Figure 5.9: $\texttt{impact}_{\texttt{SA}}$ for sender anonymity *during the transition phase* for all different adversarial strategies (from top left to bottom right): k-collusion, bandwidth, geographic, and monetary.

Transition Phase

In existing evaluations of alternative path selection algorithms, the major impeding anonymity factor that is typically omitted is the so-called transition phase, i.e., a small number of users is already using an alternative of Tor's path selection algorithm, whereas the vast majority still uses the standard one. Intuitively, the users that use the alternative algorithm cannot yet hide in a large anonymity set of users, but have to stay anonymous amongst the users that use the standard algorithm. Figure 5.10 depicts the adversary's advantage in such scenarios (for sender and relationship anonymity only, since these are the only two anonymity notions affected by this algorithmic transition).

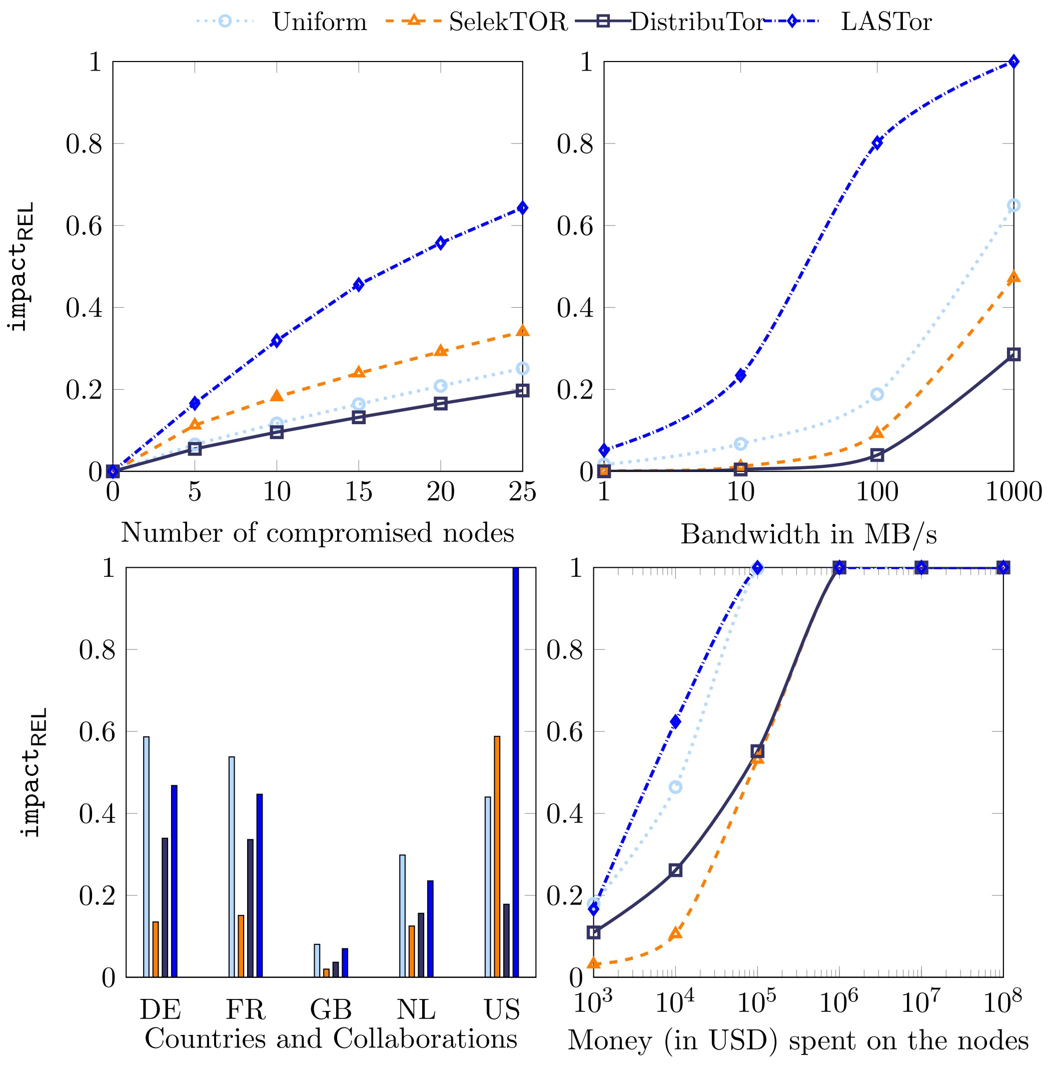


Figure 5.10: $\texttt{impact}_{\texttt{REL}}$ for relationship anonymity *during the transition phase* for all different adversarial strategies (from top left to bottom right k-collusion, bandwidth, geographic, and monetary.

Our analyses show that even adversaries that do not compromise any single Tor node have a tremendous advantage in distinguishing a user that relies on an alternative path selection algorithm from a regular Tor user: for SelekTOR, the adversary has an advantage of 90.1%, for LASTor an advantage of 86.0% and for the uniform path selection, the adversary has an advantage of 66.0%. In case of SelekTOR, this advantage arises mostly from the fact, that a normal user would choose a US exit node with only ≈ 9% probability, whereas a SelekTOR user always uses such an exit node. For LASTor and the uniform path selection algorithm, circuits containing small nodes are chosen with a much higher probability in comparison to TorPS. The effect of the

```
GreedySet($B, f, \mathcal{N}$, [impact(n) for n $\in \mathcal{N}$])
 1: Out:=$\emptyset$, $L$:=$\emptyset$
 2: for n $\in \mathcal{N}$ do
 3:   if f(n) = 0 then
 4:     Out:=Out $\cup$ {n}
 5:   else if $f$(n) $\leq B$ then
 6:     $L$:=$L \cup$ {n}
 7:   end if
 8: end for
 9: $B'$:=$B$
10: for n $\in L$, ordered by $\frac{\texttt{impact(n)}}{f(\mathsf{n})}$ (descending) do
11:   if $B' \geq f$(n) then
12:     Out:=Out $\cup$ {n}
13:     $B'$:=$B' - f$(n)
14:   end if
15: end for
16: return Out
```

Figure 5.11: Greedy algorithm for the lower bound on Tors anonymity. We instantiate impact depending on the anonymity notion. For sender anonymity, $\texttt{impact}(\mathsf{n})\texttt{:=}\texttt{impact}^{\mathcal{O}}_{\mathsf{SA}}(\mathsf{n}) + \texttt{impact}^{\texttt{ind}}_{\mathsf{SA}}(\mathsf{n}, \mathrm{A}^B_f)$, for recipient anonymity we set $\texttt{impact}(\mathsf{n})\texttt{:=}\texttt{impact}^{\mathcal{O}}_{\mathsf{RA}}(\mathsf{n}) + \texttt{impact}^{\texttt{ind}}_{\mathsf{RA}}(\mathsf{n}, \mathrm{A}^B_f)$, and for relationship anonymity we set $\texttt{impact}(\mathsf{n})\texttt{:=}\texttt{impact}^{\texttt{combined}}_{\mathsf{REL,S0,S1,R}b\mathsf{,R}d}(\mathsf{n}, \mathrm{A}^B_f) + \texttt{impact}^{\texttt{ind}}_{\mathsf{REL,S0,S1,R}b\mathsf{,R}d}(\mathsf{n}, \mathrm{A}^B_f)$.

transition phase on DistribuTor is less drastic, but still noticeable. Especially the fact that the weights of entry nodes are heavily modified grants compromised middle nodes an advantage in distinguishing between a Tor user and a DistribuTor user.

Mitigating the Risk of the Transition Phase The high vulnerability of users that use a non-standard path selection algorithm indicates that a slow and voluntary transition from one algorithm to another might alienate the (few) users that migrate first and thus significantly weaken their anonymity. We think that as soon as a transition is necessary, the novel algorithm should be rolled out to all users at once, in order to shorten the transition phase. With this strategy, all users would intuitively remain in the same anonymity set.

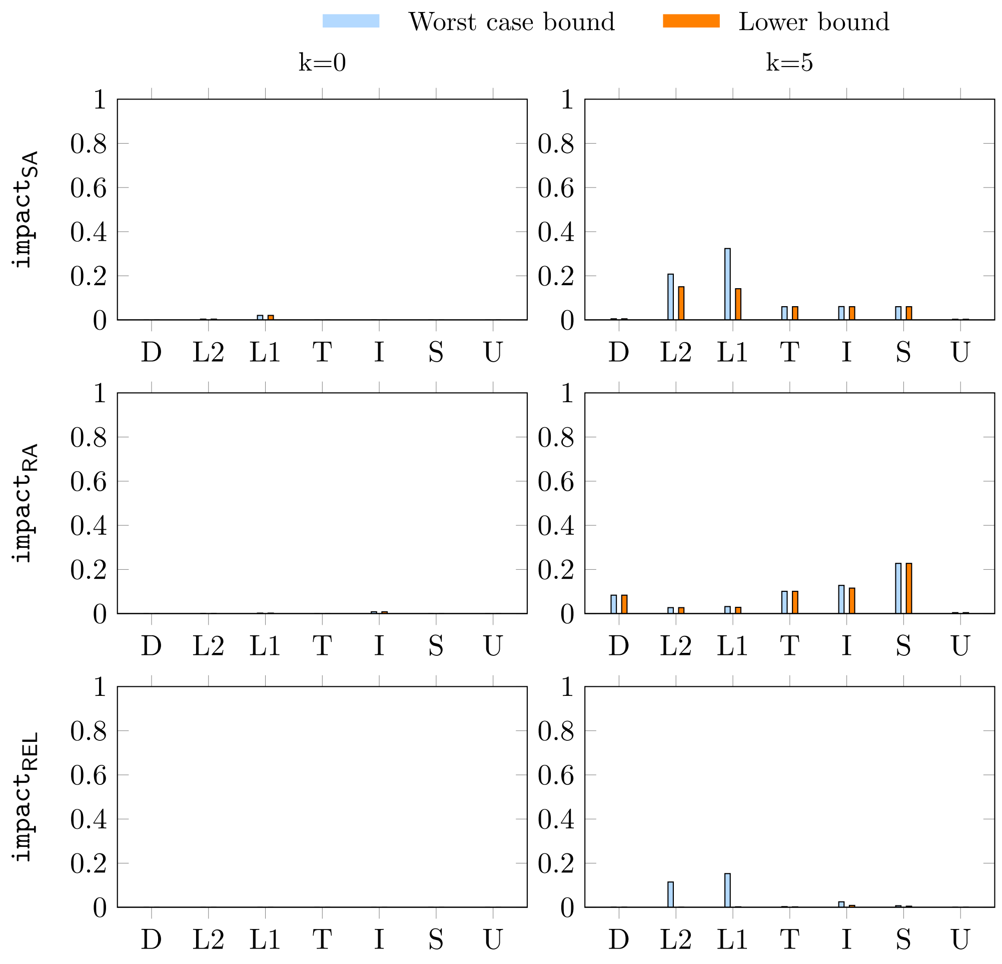


Figure 5.12: Comparison between worst-case guarantee and lower bound for: (D) DistribuTor, (L1) LASTor in our first configuration, (L2) LASTor in our second configuration, (T) Tor's current path selection algorithm, (I) Tor's current path selection algorithm if one recipient requires IRC's TCP port 6667, (S) SelekTOR for country "US", (U) Uniform. We present guarantees for sender (top), recipient (middle) and relationship (bottom) anonymity for the k-collusion adversary with k=0 (left) and k-collusion adversary with k=5 (right).

5.1.5 Evaluation of the Precision

Finally, we evaluate the precision of our results, i.e., the amount of over-approximation induced by our calculations. To this end we define a (very simple) algorithm GreedySet that greedily selects Tor nodes based on the impact per node estimated by our calculations from Section 4.3 (c.f. Fig. 5.11 for a description of GreedySet). We then calculate the precise anonymity impact of the resulting set of nodes and thus obtain a lower bound for the adversarial impact.

In Figs. 5.12 to 5.17 we compare both bounds. For sender anonymity and recipient

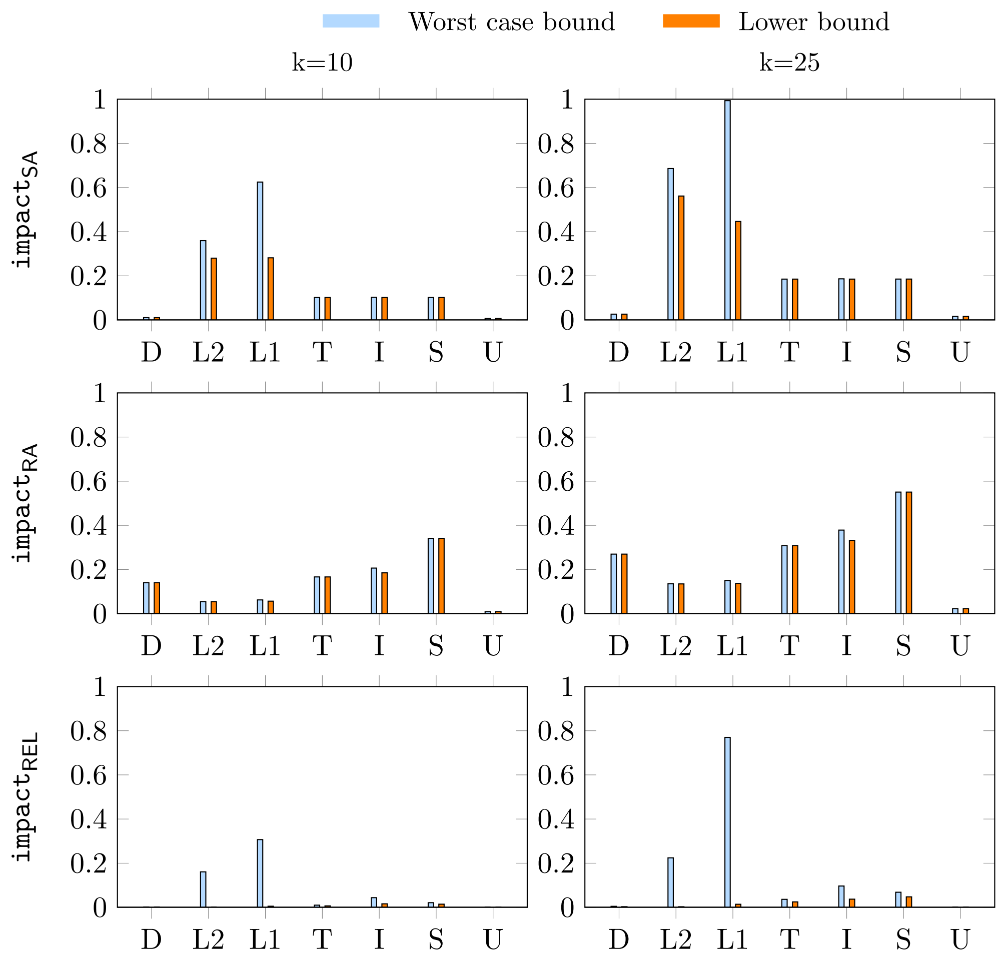


Figure 5.13: Comparison between worst-case guarantee and lower bound for: (D) DistribuTor, (L1) LASTor in our first configuration, (L2) LASTor in our second configuration, (T) Tor's current path selection algorithm, (I) Tor's current path selection algorithm if one recipient requires IRC's TCP port 6667, (S) SelekTOR for country "US", (U) Uniform. We present guarantees for sender (top), recipient (middle) and relationship (bottom) anonymity for the k-collusion adversary with k=10 (left) and k-collusion adversary with k=25 (right).

anonymity, whenever the probability distribution over Tor circuits is the same for both senders / both recipients, we do not over-approximate the adversary's advantage. Consequently, the the bound calculated by *MATor* is tight up to minor rounding errors. For relationship anonymity (where we over-approximate the impact of the adversary even in such scenarios) and for sender anonymity in cases in which the senders use different path selection algorithms, as well as for recipient anonymity in the case that different ports are required, as well as in general for LASTor, *MATor* significantly over-approximates the adversarial impact.

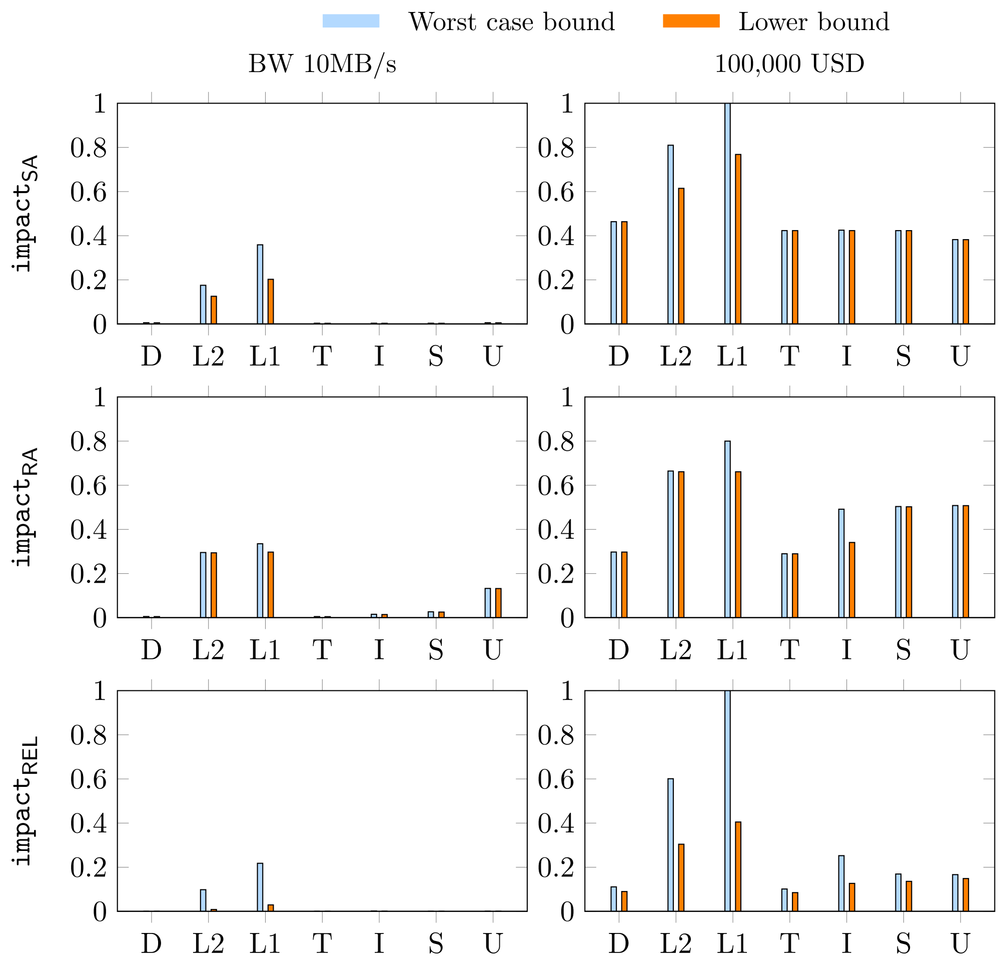


Figure 5.14: Comparison between worst-case guarantee and lower bound for: (D) DistribuTor, (L1) LASTor in our first configuration, (L2) LASTor in our second configuration, (T) Tor's current path selection algorithm, (I) Tor's current path selection algorithm if one recipient requires IRC's TCP port 6667, (S) SelekTOR for country "US", (U) Uniform. We present guarantees for sender (top), recipient (middle) and relationship (bottom) anonymity for the bandwidth adversary with B=10 MB/s (left) and the money adversary with B=100000 USD (right).

The over-approximation is more significant whenever the indirect impact plays a larger role and is most severe for relationship anonymity. Our graphs portray both the worst-case bound calculated by *MATor* and the lower bound calculated by calculating our provably tight bound for the output of GreedySet (based on the values for `impact` derived by *MATor*).

There are several sources that might lead to such a discrepancy in our bounds. Firstly, our quantification of the indirect impacts might not be as tight as possible. Secondly we might over-approximate the direct impacts by counting the same circuit

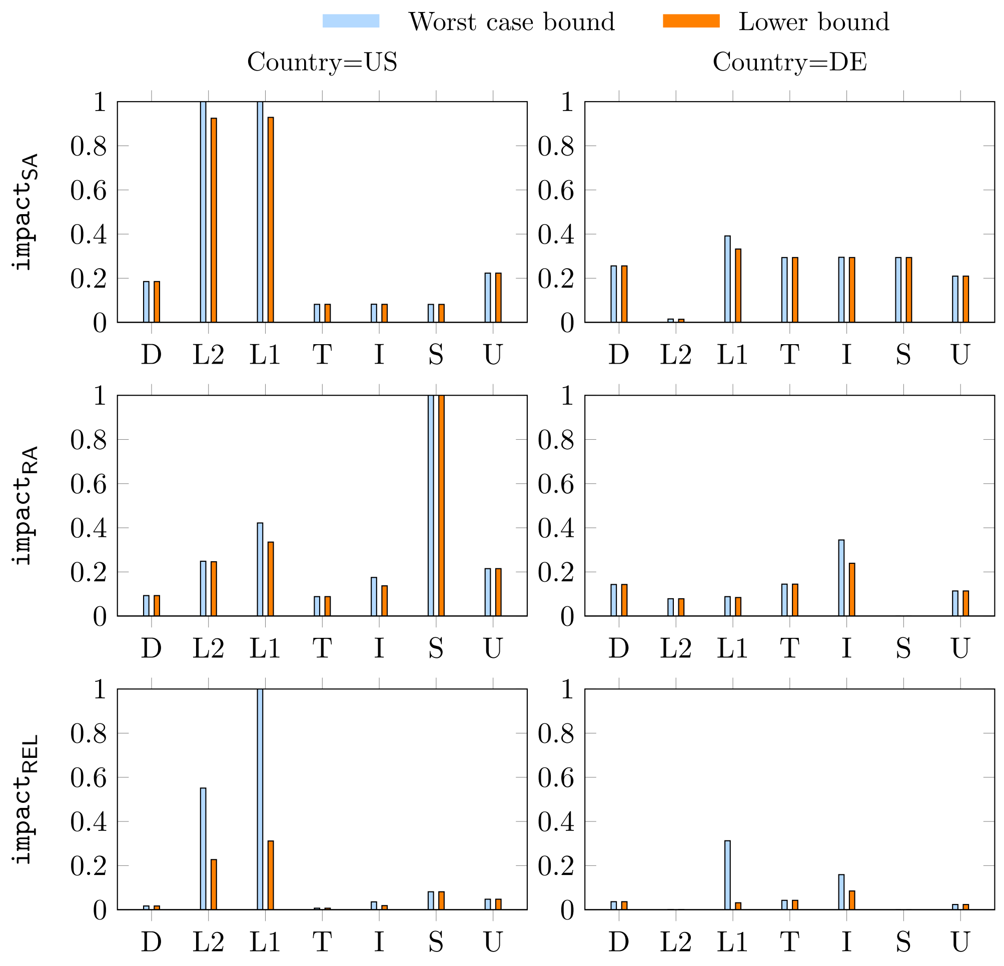


Figure 5.15: Comparison between worst-case guarantee and lower bound for: (D) DistribuTor, (L1) LASTor in our first configuration, (L2) LASTor in our second configuration, (T) Tor's current path selection algorithm, (I) Tor's current path selection algorithm if one recipient requires IRC's TCP port 6667, (S) SelekTOR for country "US", (U) Uniform. We present guarantees for sender (top), recipient (middle) and relationship (bottom) anonymity for the country adversary US (left) and the country adversary DE (right).

more than once, and lastly the algorithm GreedySet might not be optimal for cases in which pairwise impacts play a big role. In such cases, we add the impact of the pair to individual nodes, i.e., the pairwise impact of two nodes n_x and n_y is also taken into account if the adversary compromises n_x and another node n_z, even if there is no pairwise impact for n_x and n_z. Understanding the source of these discrepancies and searching for more precise guarantees as well as more precise lower bounds is left open for future work.

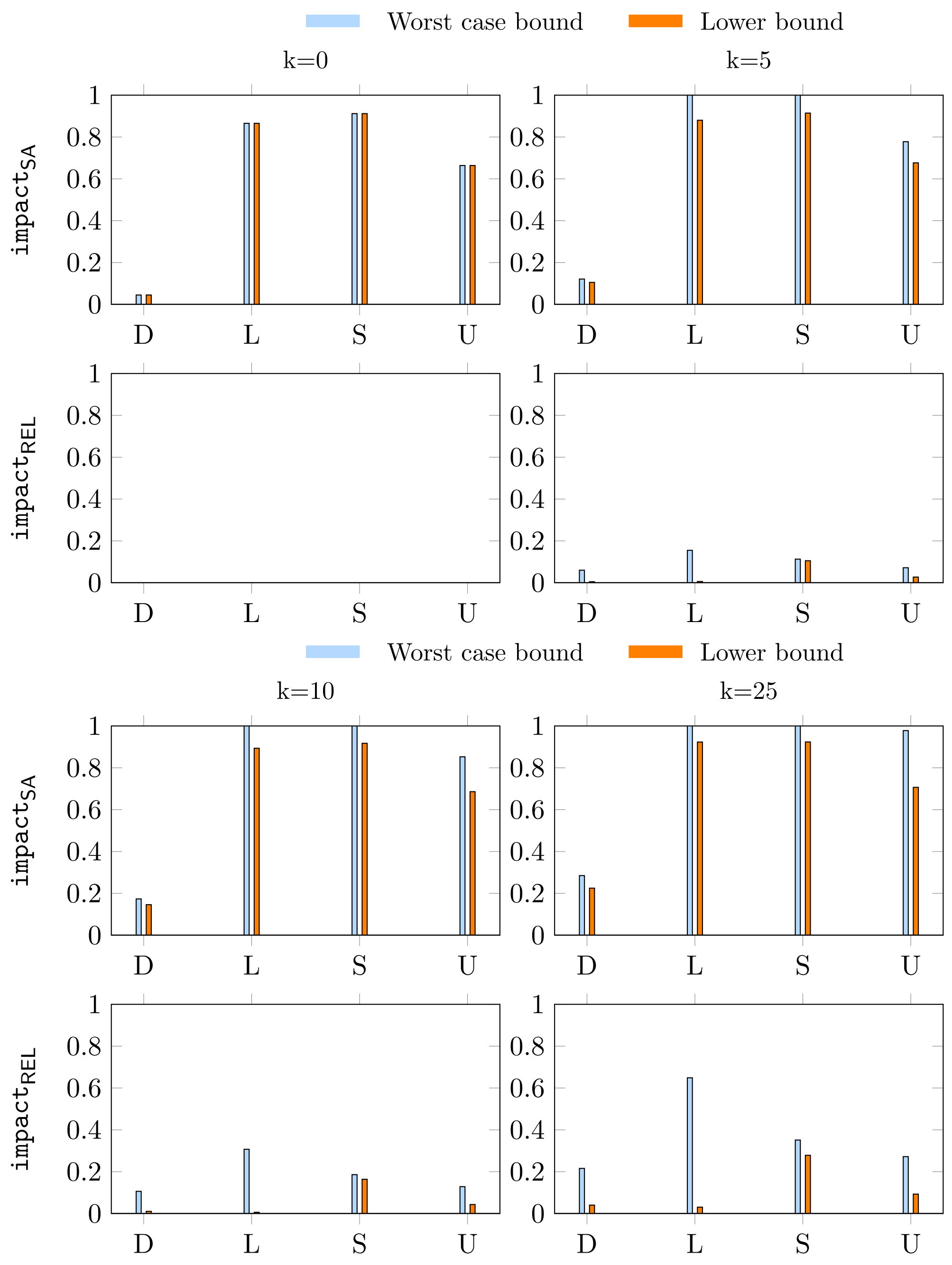
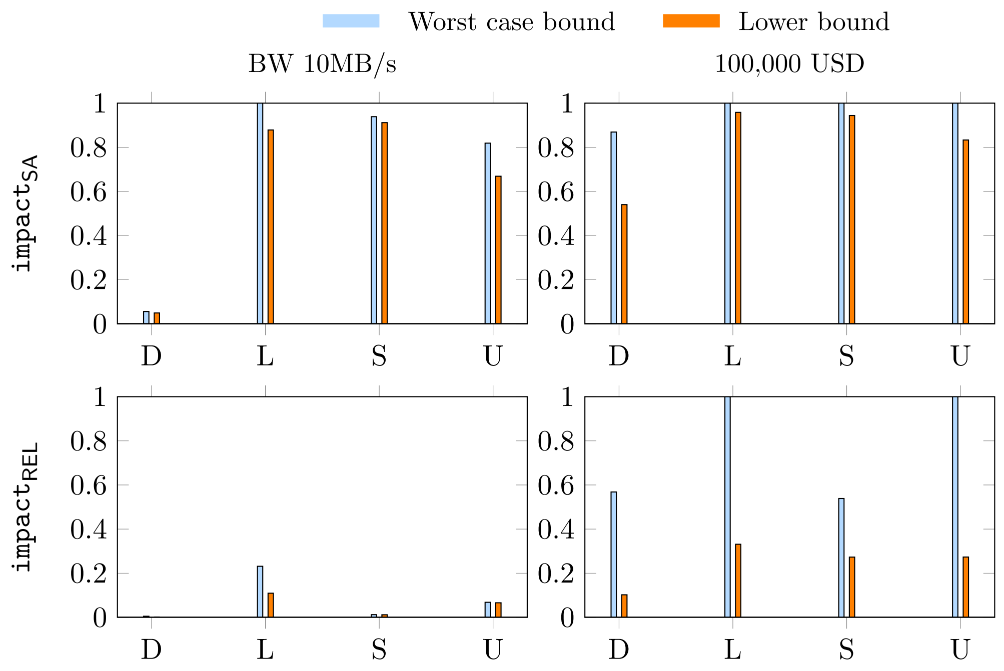


Figure 5.16: Comparison between worst-case guarantee and lower bound for for all path selection algorithms in the transition phase:(D) DistribuTor, (L) LASTor, (S) SelekTOR for country "US", (U) Uniform. We present guarantees for sender (top), recipient (middle) and relationship (bottom) anonymity for the k-collusion adversary with k=0 (top left), k=5 (top right), k=10 (bottom left) and k-collusion adversary with k=25 (bottom right).



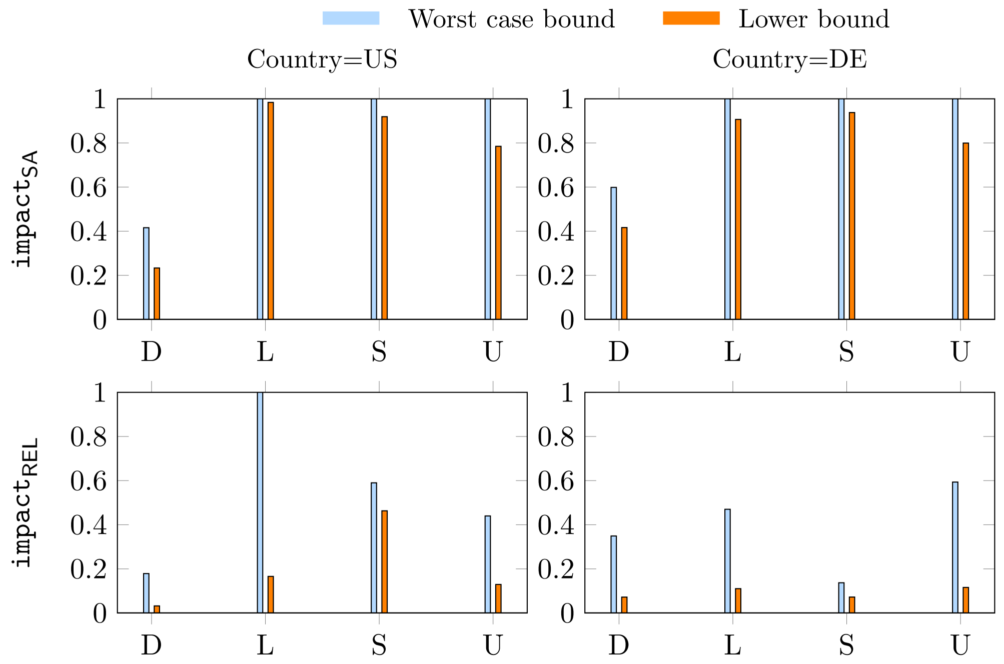


Figure 5.17: Comparison between worst-case guarantee and lower bound for for all path selection algorithms in the transition phase:(D) DistribuTor, (L) LASTor, (S) SelekTOR for country "US", (U) Uniform. We present guarantees for sender (top), recipient (middle) and relationship (bottom) anonymity for the bandwidth adversary with B=10 MB/s (top left) the money adversary with B=100000 USD (top right), the country adversary US (bottom left) and the country adversary DE (bottom right).

AS-Name	ASN	countries
LEVEL3 - Level 3 Communications	3356	21
LVLT-3549 - Level 3 Communications	3549	18
NTT-COMMUNICATIONS- NTT America	2914	17
NTTA-3946 - NTT America	3949	7

Figure 5.18: A small selection of important ASes with a wide geographic distribution.

5.2 Evaluating Tor Against Network Adversaries

In this section we instantiate the observed links between Tor entities $L \subseteq \mathcal{E} \times \mathcal{E}$ by an actual *malicious infrastructure* MI for a Tor network state (i.e., a so-called *Tor consensus*), considering real (groups of) autonomous systems (ASes), Internet exchange points (IXPs) and submarine cables.

To properly define the impact of malicious infrastructure, we construct a model of Tor's Internet topology that consists of all relevant *routing paths* between Tor entities. A routing path between two Tor entities n_1 and n_2 is a set of ASes, IXPs and submarine cables that are able to observe traffic between n_1 and n_2.

5.2.1 Internet Topology Datasets

Before we describe our model of Tor's Internet topology, we briefly discuss two possible directions for building such a model – publicly available Border Gateway Protocol data published by several organizations and the iPlane project – and compare them in terms of accuracy and coverage.

BGP: The Border Gateway Protocol (BGP) is the standard protocol for negotiating and announcing Internet routing between different ASes. In BGP, an AS announces to its peering ASes to which IP prefixes it can connect them, and these peers can relay this information to their respective peers. BGP data published by an AS consequently contains information about the paths that traffic will most likely take from exactly this AS to the announced IP prefixes. Since several organizations provide free BGP data feeds [112, 28], it is tempting to leverage BGP data for constructing a model of Internet routing. However, these local views on the Internet are each specific to one AS. To construct an accurate model of Tor's Internet topology, we would require BGP information from respective ASes of a large fraction of Tor nodes. Moreover, BGP information is very coarse-grained, as it only reveals the ASes on the route, but not the precise routes. Hence it lacks the geographical precision necessary for Internet exchange points and submarine cables, as many, especially the often-used Tier-1 ASes span dozens of countries and even several continents. We refer to Figure 5.18 for a small collection of such ASes. Consequently, we consider BGP data to be insufficient for our model.

iPlane: The iPlane project [76] combines active traceroute measurements, which are periodically performed between nodes under control of the project (e.g., PlanetLab nodes), with opportunistic measurements to achieve a wider, but still precise coverage of today's Internet topology. To increase the coverage of its traceroute information, iPlane clusters IP addresses on the basis of BGP atoms (i.e., IP prefixes), and treats hosts at the same point of presence (PoP) as equal. Furthermore, as an opportunistic method, iPlane predicts paths based on known sub-paths, applying several sanity checks to them to ensure precision. Technically, iPlane supplies IP routes as well as AS routes between tuples via an API [76]. Consequently, iPlane achieves a significantly wider coverage than individual traceroute measurements, while maintaining a higher accuracy than BGP data sources. The precision of iPlane is in the granularity of IP prefixes, and, often, even as fine-grained as IPs, which allows for a quite precise geographic localization of the involved systems and thus for modeling IXPs and submarine cables. Consequently, we utilize iPlane for constructing our model of Tor's Internet topology.

Tor-to-Tor Traffic

We construct a model of all routing paths between Tor nodes, by first selecting a Tor consensus (from 2015, June 25th) and subsequently querying iPlane with all pairs of IP prefixes that contain at least one Tor node from this consensus. In turn, iPlane provides a large quantity of routing paths between Tor nodes, but only reaches a coverage of around 24% for the considered consensus. We then increase our coverage by restricting the consensus to the Tor nodes for which we have routing path information. This restriction slightly reduces the significance of our evaluation, but increases the precision of our calculation.

To this end, we propose a graph-based algorithm that reduces the considered Tor consensus in order to achieve high connectivity; here we define the connectivity of a Tor node n_1 as the number of other Tor nodes n_2, such that the path from n_1 to n_2 is within our dataset. Figure 5.19 describes our greedy algorithm for finding a good trade-off between connectivity and significance by successively shrinking the consensus until the average connectivity ratio of all remaining nodes is above a specified target threshold. To this end, we select the node(s) with the smallest connectivity as potential candidates for removal. To increase the significance of our result, we add all nodes whose connectivity is below a smoothed threshold of $1 + s$ times the connectivity of the least connected node. We then remove the candidate node with the smallest significance to the consensus (i.e., the smallest bandwidth weight) from the consensus. The intuition behind our algorithm's smoothness is that we prefer to remove slightly better connected nodes, if their impact to the consensus is smaller than that of slightly less connected node. We hence optimize the consensus towards a significant amount of remaining bandwidth weight. We run the algorithm several times with different smoothness values s to obtain an ideal result.

We use three metrics to evaluate the efficacy of our algorithm from Figure 5.19,

```
Reduce Consensus
  For all pairs of Tor nodes $\mathsf{n}_1, \mathsf{n}_2$, let $\mathit{info}(\mathsf{n}_1, \mathsf{n}_2) \in \{0,1\}$ be 1 iff we have routing path information from $\mathsf{n}_1$ to $\mathsf{n}_2$.
  Let $\mathit{conn}_{\mathsf{n}}$ be the connectivity of node $\mathsf{n}$ for all nodes.
  Let $L := \mathcal{N}$ be the list of nodes we consider (initially all).
  while $\mathit{average}_{\mathsf{n} \in L}\,(\mathit{conn}_L(\mathsf{n})) < \mathit{goal}$ do
    $\mathit{worst} := \mathit{min}_{\mathsf{n} \in L}\,(\mathit{conn}_L(\mathsf{n}))$.
    Let $\mathit{candidates} := \{\mathsf{n} \in L \; s.t. \; \mathit{conn}_L(\mathsf{n}) \approx \mathit{worst}\}$
    Let $r \in \mathit{candidates}$ s.t. $r.\mathit{bandwidth}$ is minimal.
    $L := L \setminus \{r\}$
  end while
```

Figure 5.19: Our algorithm for greedily reducing a given Tor consensus with emphasis on high connectivity based on available routing path data as well as optimizing towards high bandwidth weights.

and compare the resulting statistics to the original consensus: (1) the distribution of Tor nodes with relevant *flags* (such as *Guard, Exit, Fast, Stable*), (2) the distribution over the Top countries, and (3) the remaining number and overall bandwidth of Tor nodes. Figure 5.20 shows the distribution of node types in the reduced consensus, depending on the target connectivity of our algorithm from Figure 5.19. We aim at very high connectivity values to increase the accuracy of our calculation. We present the distributions over node types for target connectivities ranging from 90% up to 99.5% in steps of 0.5%. Note that the distribution over Flags does not significantly change with the target connectivity. Moreover, Figure 5.20 demonstrates that also the distribution of Tor nodes per country is relatively static and deviates at most 5% from the original consensus, up to connectivity values of 99%. Finally, Figure 5.20 shows the total Tor bandwidth of the remaining nodes and the number of remaining nodes, as a percentage of the original Tor consensus. Given the significant drop of bandwidth beyond 0.97% coverage, we chose to fix 0.97% coverage as a parameter for Figure 5.19. The reduced consensus we considered includes 1677 Tor nodes out of 6680 ($\approx$ 25.1%) with a total bandwidth weight of 15.11 Mio. out of 34.82 Mio ($\approx$ 43.39%). Up to this point, our model of Tor's Internet topology is state-of-the-art both in terms of coverage and precision of the routing paths between Tor nodes and in terms of significance.

Tor-to-User Traffic

We chose to select IP addresses as senders and recipients from which we can immediately run traceroute measurements to all relevant Tor nodes, thus ensuring a perfect connectivity and a high precision out our routing paths into and out of the Tor network. We refer to Section 5.2.4 for our choice of senders and recipients.

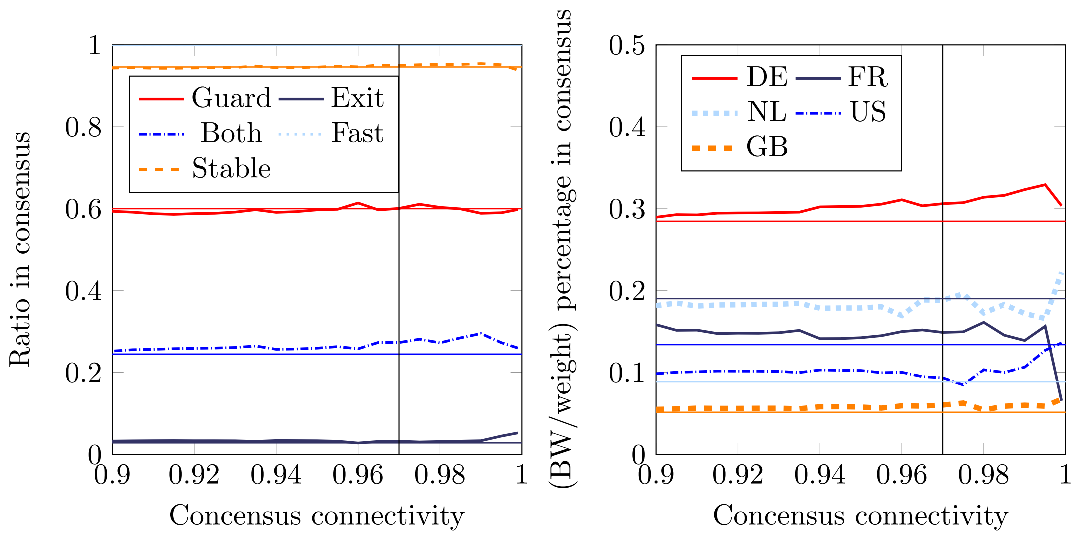


Figure 5.20: **Reduced Consensus:** In this figure we present both the Tor node type distribution per threshold (left) and the Tor node country distribution per threshold (right). The horizontal lines show the original ratio, the dashed lines the ratio of the reduced concensus. The vertical line marks the properties of the consensus we chose for our evaluation.

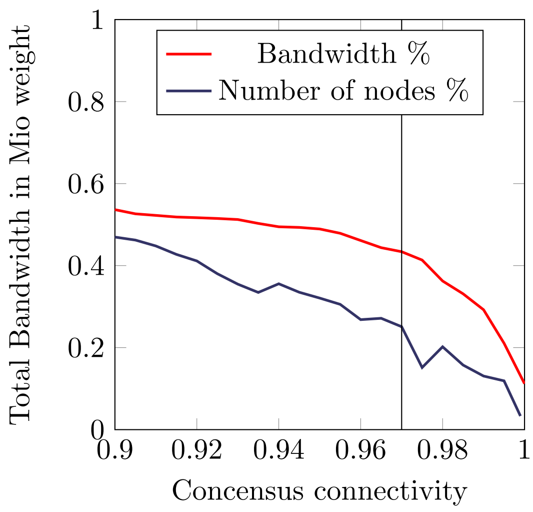


Figure 5.21: The size of the remaining bandwidth and number of nodes of the reduced Tor consensus, depending on the threshold. The values are given in the percentual coverage, where 100% marks the value of the original consensus.

5.2.2 Network-level Adversaries

The precise topology information in terms of routing paths allows us to assess the anonymity reduction achieved by various network-level adversaries. We focus on those adversarial models that we will use in our evaluation: *(combinations of) ASes*, *Internet exchange points* and *submarine cables.* In general, our framework can be extended to many other types of adversaries, as long as we can model the set of routes that an adversary can monitor. That is, we have to model which ASes or other types of intermediates along the route an adversary can monitor. As we intent to analyze the impact of eavesdroppers on Tor, we assume that the adversary does not deploy active attacks on the network level (such as traffic manipulation, BGP hijacking, etc.).

Autonomous Systems (AS)

As the simplest adversary model, we consider an AS-level attacker. That is, we model an attacker that can monitor all traffic that origins in an AS, destines in an AS, or passes an AS as part of a route path between senders and Tor guard nodes, between two Tor nodes or between Tor exit nodes and recipients. We translate the route paths generated by iPlane to IP prefixes, by applying a mapping from the Routeviews project[96]. Subsequently, we translate these prefixes to ASes, by comparing them to the list of announced prefixes.

Internet Exchange Points (IXP)

As a more sophisticated adversary we consider malicious Internet Exchange points (IXPs). IXPs have been established to allow peering between providers in a star topology, i.e., as a (typically economically more attractive) alternative to using services of upstream providers. However, in contrast to ASes, an IXP typically operates on a link layer and thus is not visible in routing paths. To model whether traffic passes an IXP, we thus use the following conservative estimation. We extract the list of ASes that are members of IXP and particularly focus on the ones with an *open* peering policy, as this policy indicates a high likelihood that traffic between two ASes traverses the IXP. In contrast, we ignore ASes with a non-open policy, and thus slightly underestimate the power of an IXP, as we cannot estimate whether such ASes have any commercial peering agreement with other ASes. For each routing path, we then check whether two consecutive IP addresses that geographically are not too far apart from the IXP are in the list of openly peering ASes, and if so, assume that the IXP was used along the route. This strategy reflects that ASes typically favorize to establish strong connections to a closeby IXP over establishing direct connections to a large number of other ASes.

Submarine Cables

Finally, we model an adversary that can passively monitor a submarine cable. We assume that a submarine cable on a routing path if both IP prefixes have a distance of at least 1000 km and if the IP prefixes are at most 500 km away from some landing point of that cable. We chose a radius of 500 km to account for imprecise GeoIP data of some IP prefixes, while still excluding other, different cables.

Geographic locations: For all mentioned geographic considerations, both of IXPs and of submarine cables, we utilized the up-to-date GeoIP2-City database of MaxMind [79], that achieves an accuracy of more than 80% in most countries worldwide.

5.2.3 Implementation and Optimizations

We implement ComputeDelta as an extension of the MATor [14] tool. MATor already computes the probability distribution over Tor circuits for a given Tor network consensus and the respective server descriptors. We extend these basic computations by means of a C++ implementation that computes ComputeDelta from Chapter 4, based on a description of Tor's Internet topology and routing path information from the considered senders to all Tor guard nodes and from all Tor exit nodes to the considered recipients. A naïve implementation of ComputeDelta would result in a prohibitively high memory consumption. We hence group observations into four groups depending on their position. Then for each group we reorder the three loops (over guard nodes, middle nodes and exit nodes) in order to directly compute the impact of each observation on the anonymity reduction. We thereby reduce memory consumption significantly, while increasing the computation time by a factor of four. We expect that our implementation can be further optimized, e.g., by leveraging massive parallelization via the use of GPU libraries such as OpenCL.

5.2.4 Senders and Recipients

Routing path information from senders to Tor guards and vice versa, as well as from Tor exit nodes to recipients and vice versa, is crucial for accurately calculating the anonymity reduction against malicious infrastructure. Consequently, we select senders and recipients in a way that allows us to derive routing path information ourselves. To this end, we use looking glass servers – a collective of different servers, distributed over the world, that provide an open HTTP interface for running Traceroute. We use the server list provided at `http://www.traceroute.org/` as a reference for available looking glass servers, and we selected servers with a small response-time and high availability from this list. We then automatically query these servers with traceroute request to all Tor guard nodes and Tor exit nodes via HTTP-requests and extract the Traceroute path from the resulting page. From all 290 working Looking Glass servers we utilized, we choose 20 stable and responsive servers with ideal coverage in different IP

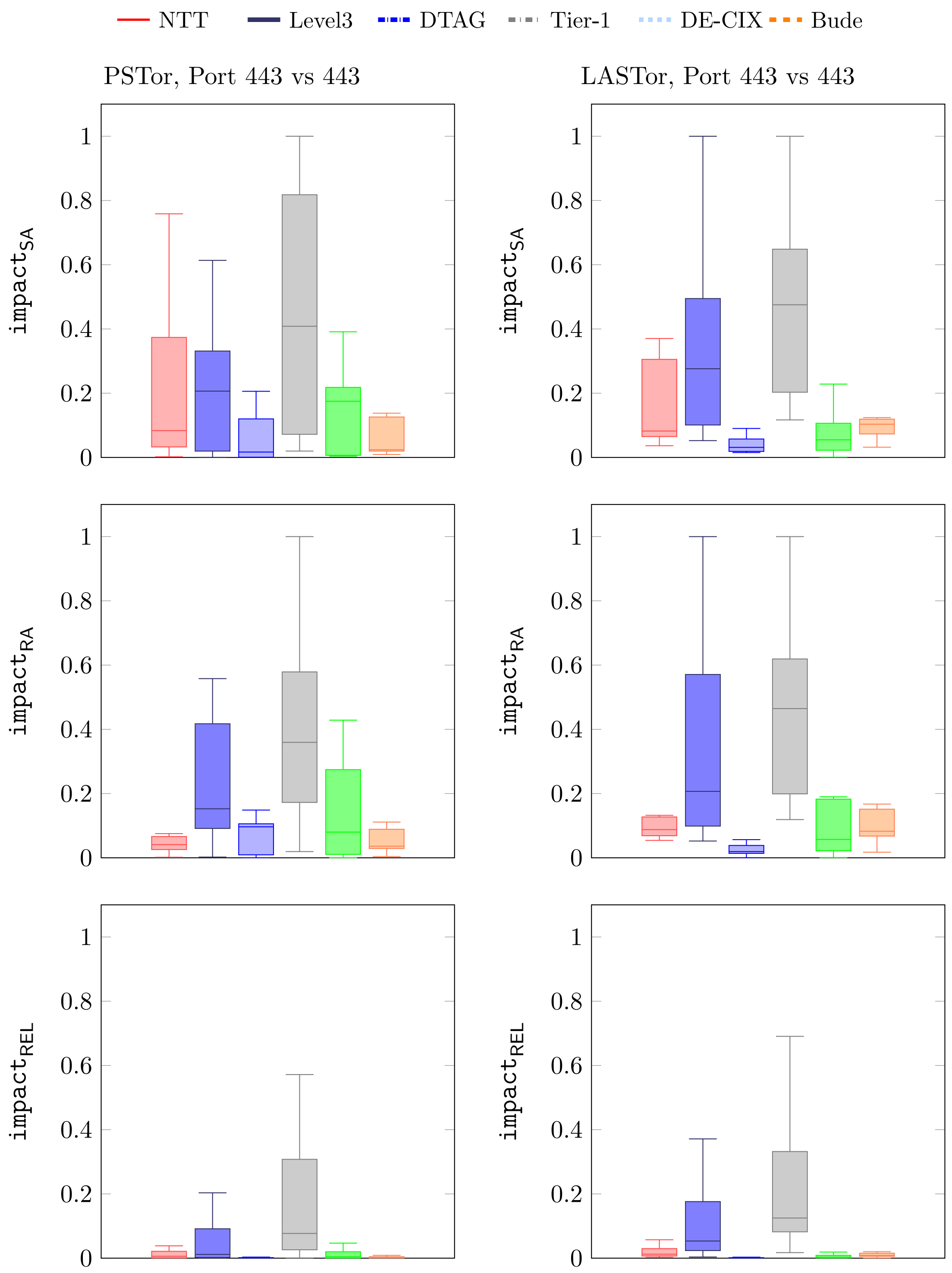


Figure 5.22: Anonymity reduction as box plots for Tor's current path selection algorithm (left) and for LASTor (right), including lower quartile, median, upper quartile and minimal and maximal values of anonymity reduction for all three anonymity notions: sender anonymity (top), recipient anonymity (middle) and relationship anonymity (bottom). The adversaries are, from left to right: NTT, level3, DTAG, Tier-1, DE-CIX, Bude

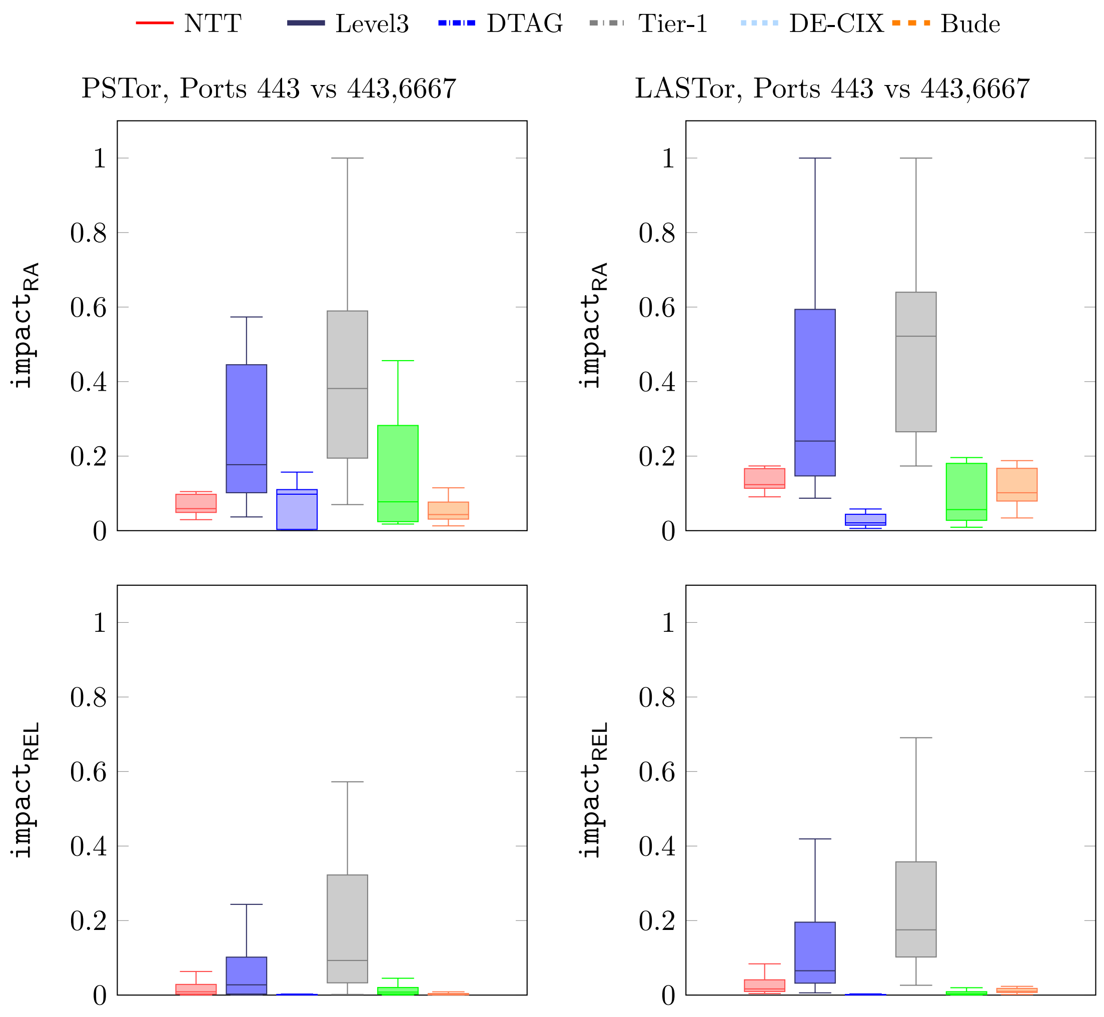


Figure 5.23: Anonymity reduction as box plots for Tor's current path selection algorithm (left) and for LASTor (right) if (in both cases) we compare a recipient that requires the TCP port 443 with a recipient that additionally requires the TCP port 6667 for all three anonymity notions: sender anonymity (top), recipient anonymity (middle) and relationship anonymity (bottom). The adversaries are, from left to right: NTT, level3, DTAG, Tier-1, DE-CIX, Bude

prefixes and select them as senders and recipients. We identified them using a tcpdump on a well-connected server under our control that captured any UDP, ICMP, and TCP traffic, and then ran individual traceroutes to our server from the used looking glass servers. We then identify the corresponding ASN via the whois service provided by team cymru and locate their geographical position via the commercial GeoIP database of MaxMind [79]. We refer to Figure 5.24 for a complete list of the selected servers, together with their geographical distribution over countries, their IP addresses, their

IP	Country	ASN	Provider
134.96.225.228	DE	680	DFN
130.206.245.93	ES	766	REDIRIS Entidad Publica Empresarial Red.es
202.10.15.153	AU	2764	AAPT AAPT Limited
213.200.64.94	DE	3257	Tinet SpA
67.16.148.38	US	3549	Level 3 Communications
137.192.48.106	US	5006	Onvoy
87.245.229.126	GB	9002	RETN Limited
212.101.0.106	CH	9044	SOLNET BSE Software GmbH
59.106.1.141	JP	9370	SAKURA Internet Inc.
210.224.179.143	JP	9371	SAKURA Internet Inc.
211.14.3.123	JP	9607	BroadBand Tower Inc.
216.182.1.7	US	10484	H&R Block
212.19.45.36	DE	12306	Plus.Line AG
213.136.1.6	NL	12859	BIT BV
213.91.2.19	FR	13273	Open Wide Outsourcing SAS
217.20.161.38	UA	15772	WNET Ukraine
81.31.41.18	CZ	24917	Master Internet s.r.o.
217.24.81.10	FR	28855	GalacSYS
82.119.252.70	CZ	29208	Dial Telecom a.s.
212.24.145.50	CZ	29208	Dial Telecom a.s.

Figure 5.24: Looking glass servers used as senders and recipients in the evaluation.

AS number and their respective providers. For each evaluation, we sampled two senders and two recipients from our 20 looking glass servers uniformly at random.

5.2.5 Evaluated Malicious Infrastructure

We now instantiate the malicious infrastructure of our analysis with actual companies and locations of interest. For ease of presentation and to achieve comparability of our

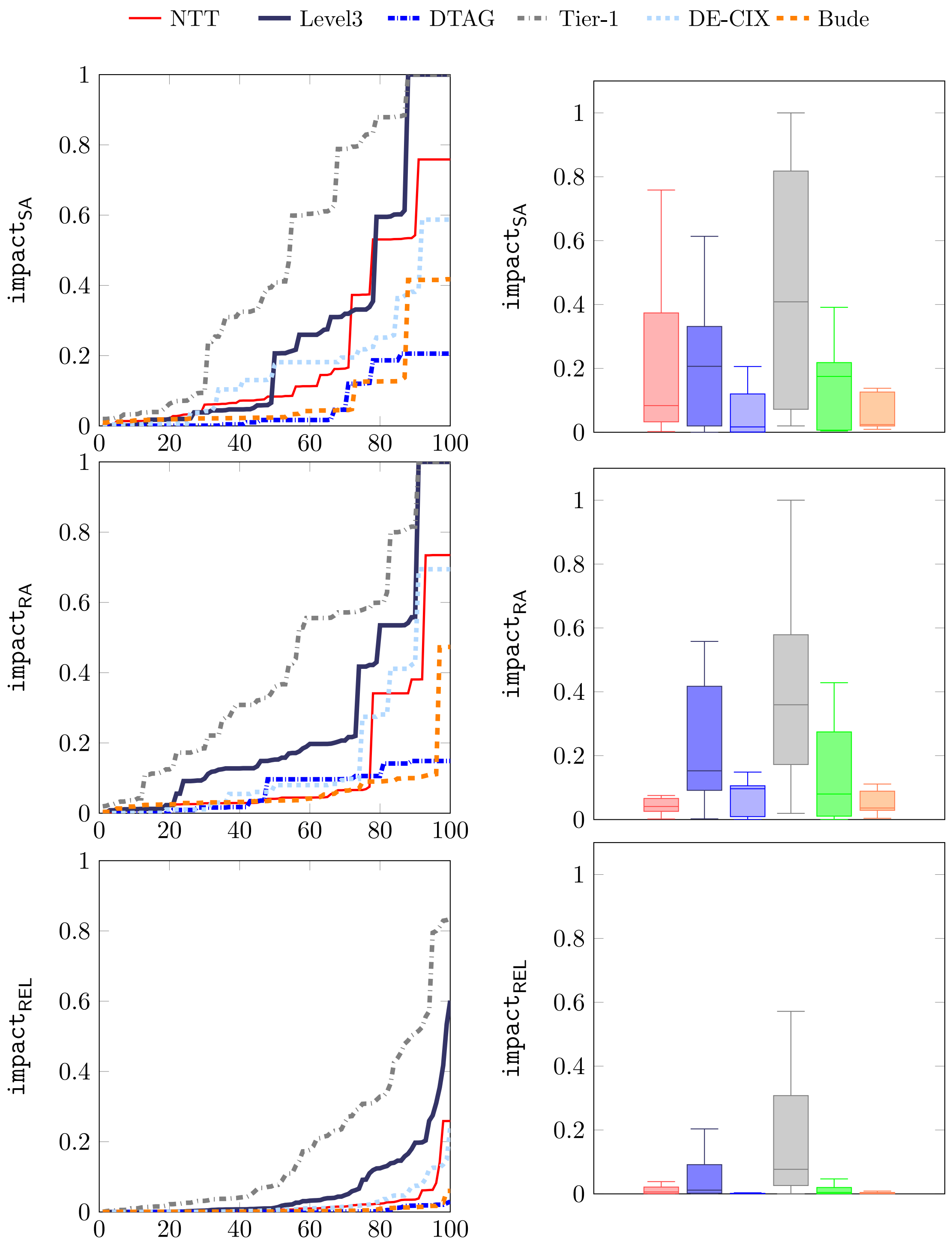


Figure 5.25: **PSTor:** Anonymity reduction per sample of senders and recipients (left), as well as box plots for these values, including lower quartile, median, upper quartile and minimal and maximal values of anonymity reduction (right) of Tor's path selection algorithm for all three anonymity notions: sender anonymity (top), recipient anonymity (middle) and relationship anonymity (bottom).

results we focus on the following six instances of malicious infrastructure: a selection of three Tier-1 providers, as well as their combination; DE-CIX as the world's largest IXP; and one of the most important landing points of submarine cables at Bude, UK.

Nippon Telegraph and Telephone (NTT): All ASes belonging to the Japan-based Tier-1 provider NTT, which has more than 20 ASes, connecting Asia, North America and Europe.

Level 3 Communications: All 15 ASes belonging to the US-based Tier-1 provider Level 3 Communications, which operates in Africa, Asia, Europe, Middle East and South America.

Deutsche Telekom AG (DTAG): All 3 ASes belonging to the Germany-based Tier-1 provider DTAG, which operates in Africa, Asia, Europe, North America, and South America.

Tier-1: All ASes belonging to Level 3, DTAG and NTT.

DE-CIX: The world's largest IXP, the German DE-CIX in Frankfurt. The DE-CIX connects more than 626 ASes, with a strong focus on Europe. As a geographic constraint we only add DE-CIX to a routing path between IP prefixes of openly peering ASes within Europe.

Bude landing point: The landing point of submarine cables at Bude, UK. Hence the adversary can observe all communication that traverses this landing point. Several submarine cables are crossing through the Bude landing point: the transatlantic cables TAT-14, Apollo, and Yellow, the Euro-African-Indian cable Europe India Gateway, the France-Ireland-England cable FastnetConnect, the England-Westafrican cable Glo-1, and the England-Ireland cable Pan European Crossing.

5.2.6 Results

For our evaluation against the aforementioned scenarios of malicious infrastructure, we first consider Tor's default path selection algorithm and considered recipients that are only contacted via HTTPS (TCP port 443). Recall that this corresponds to the main usage of Tor, as the Tor-Browser applies HTTPS-Everywhere. We thus consider senders and recipients for which the distribution over Tor circuits is equal, so that an adversary can gain information by observing traffic into the Tor network (for sender anonymity) or out of the Tor network (for recipient anonymity) or both (for relationship anonymity).

Our results for Tor's default path selection algorithm are given in Figure 5.25 as three graphs, for sender anonymity, recipient anonymity, and relationship anonymity. In each graph, one line corresponds to one scenario of malicious infrastructure, showing all samples of sender/recipient pairs (x-axis) ordered by the respective anonymity reduction δ (y-axis) according to Definition 2.2.1. For ease of comparison, we added box plots for these graphs (Figure 5.25, lower part), presenting the lower quartile, the inner quartile (median) and the upper quartile, with whiskers showing the lowest and highest sample (excluding outliers with more than 1.5 times the difference between upper quartile and lower quartile, which rarely occurred).

In Figure 5.26 we then compare our results for Tor's default path selection algorithm with our results for LASTor, depicting the difference between LASTors anonymity reduction and Tor's default anonymity reduction (per sample) for each scenario of malicious infrastructure and the absolute values of our results for LASTor.

Sender anonymity

For sender anonymity we considered a scenario where the sender visits a malicious website, i.e., connects to a malicious recipient. Consequently, the adversary can (in addition to the observations using malicious infrastructure) observe traffic from the Tor network to the recipient.

Tor: All considered adversaries noticeably reduce sender anonymity. Both Level 3 and NTT reduce sender anonymity significantly, for the majority of samples. The DE-CIX adversary is geographically restricted, and thus its success depends mainly on the location of the senders. In particular, DE-CIX reduces the anonymity of senders residing in Europe by more than 17% on average, while reducing anonymity by less than 1% for senders outside of Europe.

A corruption of the Bude landing point results in a reduction of sender anonymity ranging from 1% up to 40%, depending on the considered sample. We attribute these results to the fact that many Tor circuits need to use some submarine cable, as Tor circuits often span more than one continent. However, since we only considered a corruption of the Bude landing point, many other cables using different landing points exist.

LASTor: LASTor clearly improves anonymity against the DE-CIX adversary. However, the Level 3 adversary, and the Bude landing point adversary become significantly stronger. We interpret this as follows. The probability to choose a guard for which DE-CIX, or, for several samples, NTT or DTAG observes S1-G is drastically reduced by LASTor. We see two possible reasons for the increased strength of the Level 3 adversary and the Bude adversary: first, the probability to use a guard to which they observe a communication from the sender is increased; and second, internal observations made within the Tor network give away information about the senders' location, e.g., in many cases a cable would be used by one sender, but not by another sender, even for traffic between the guard node and the middle node. Consequently, observations made within a Tor circuit can help in distinguishing between two senders. Although such an observation does not completely deanonymize the sender, it gives away partial information, and it substantiates the necessity to cover the more subtle information that an adversary can exploit. We attribute the overall increase of DTAG and NTT to the second reason as well.

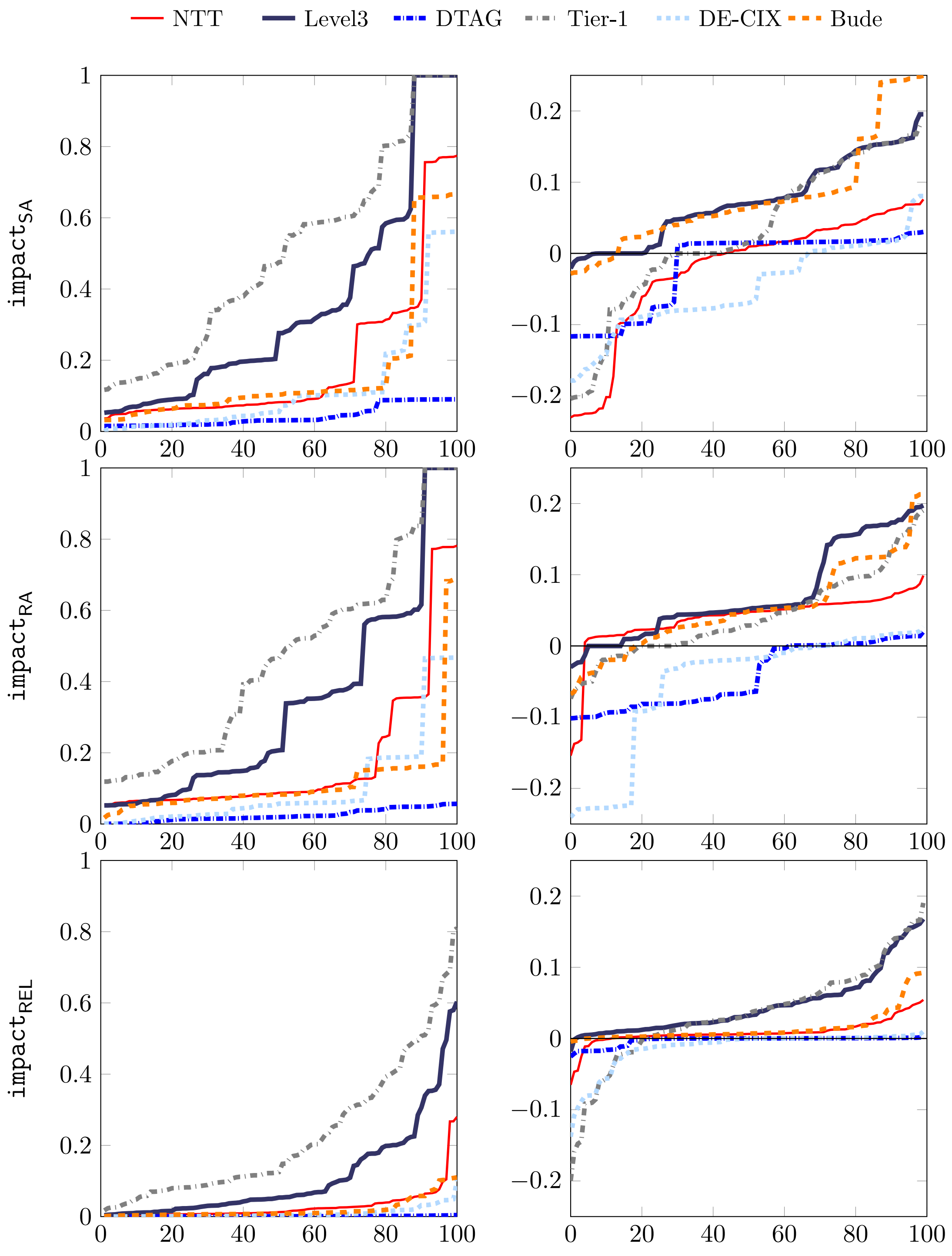


Figure 5.26: **LASTor:** Our results for LASTor per sample of senders and recipients for all three anonymity notions: sender anonymity (top), recipient anonymity (middle) and relationship anonymity (bottom). We plot both the impact of the adversaries on LASTor (left) and the difference of their impact in comparison with Tor's current path selection algorithm (right). Note that the y-axis for our differences ranges from -.25 to .25. A higher value means that LASTor offers in these cases less anonymity compared to the setting in Figure 5.25.

Recipient anonymity

For recipient anonymity we considered a scenario where the adversary can observe the Internet connection of the sender. Consequently, the adversary can (in addition to the observations using malicious infrastructure) observe traffic from the sender into the Tor network. As for sender anonymity, all considered adversaries noticeably reduce recipient anonymity. Level3 reduces anonymity more significantly than for sender anonymity: by over 10% for 70% of our samples. We observe that for all considered scenarios of malicious infrastructure, the variance of the anonymity reduction is significantly smaller for recipient anonymity than for sender anonymity. In particular, the anonymity reduction depends less on the positions of sender and recipient, and more on the positions of Tor's exit nodes. We attribute this dependency to the smaller number of Tor exit nodes.

Relationship anonymity

The relationship anonymity reduction of the considered malicious infrastructure is naturally smaller than for sender anonymity and for recipient anonymity, since relationship anonymity is inherently harder to break. Consequently, most considered scenarios of malicious infrastructures only reduced relationship anonymity by a small degree, with the noteworthy exception of the Tier-1 provider Level 3. Note that the combination of our three Tier-1 providers in comparison is extremely strong. For relationship anonymity, their combined impact is significantly higher than the sum of the impacts of the individual providers and reaches values above 20% for almost half of our samples. These results show that large (groups of) autonomous systems pose a significant threat for relationship anonymity, and thus, for every usage of Tor. Their impact is even stronger against LASTor, as even in cases in which these groups cannot completely deanonymize a connection, they can often gain partial information about the geographic location of senders and recipients.

5.2.7 Different TCP Ports

So far we analyzed Tor and LASTor for HTTPS only, which corresponds to the main usage of Tor. However, Tor can be used for (almost) arbitrary TCP traffic. For the following analyses, we hence compare two recipients with different requirements: one is only contacted via HTTPS on port 443; the other one provides an Internet Relay Chat (IRC) and thus additionally requires the port 6667. As this port is not supported by all Tor exit nodes, it restricts the path selection algorithm. We present the anonymity reduction imposed by the considered scenarios of malicious infrastructure in Figure 5.27 (both for Tor's path selection algorithm for LASTor). Considering ports typically increases the anonymity reduction. However, in contrast to what we have observed for compromised Tor nodes in Section 5.1 (where restricting the path selection algorithm for S_0–R_1, but not for S_0–R_0, always reduces anonymity), we can identify cases in which

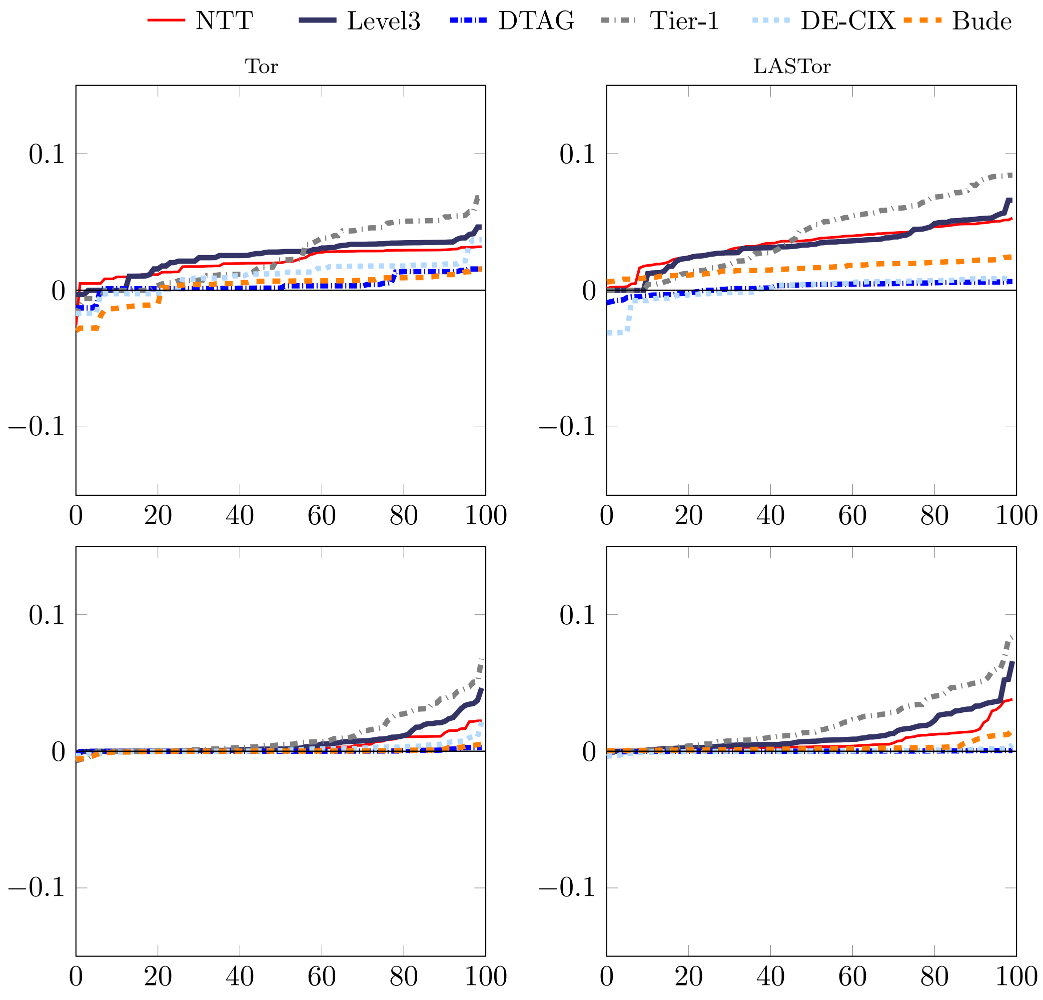


Figure 5.27: **Difference Distinct Ports PSTor and LASTor:** Impact of using different TCP ports on the anonymity reduction per sample of senders and recipients of Tor's path selection algorithm (left) and LASTor (right) for recipient anonymity (top) and relationship anonymity (bottom). Note that the y-axis ranges from -0.15 to 0.15. We compare a user requesting port 443 (HTTPS) with a user requesting port 443 and 6667 (IRC). A higher value means that in these cases different ports with PSTor/LASTor leads to less anonymity compared to the setting in Figures 5.25 and 5.26.

the anonymity reduction is less severe if one recipient requires a less supported port, like IRC (see, e.g., the leftmost results for Bude in Figure 5.27). For instance, consider a heavy-weight exit node X that only supports port 443 and an adversary that observes the connection between X and R_1, but not between X and R_0. While the adversary can deanonymize the recipient if it observes traffic from X to R_1, no such observation can be made anymore if R_1 requires a different port, since X cannot be selected as an exit node.

Bibliography

[1] M. Akhoondi, C. Yu, and H. V. Madhyastha. “LASTor: A Low-Latency AS-Aware Tor Client”. In: *Proc. of the 2012 IEEE Symposium on Security and Privacy (S& P)*. IEEE Computer Society, 2012, pp. 476–490.

[2] M. AlSabah, K. Bauer, and I. Goldberg. “Enhancing Tor’s performance using real-time traffic classification”. In: *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM. 2012, pp. 73–84.

[3] M. AlSabah and I. Goldberg. “PCTCP: per-circuit TCP-over-IPsec transport for anonymous communication overlay networks”. In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM. 2013, pp. 349–360.

[4] M. AlSabah, K. Bauer, I. Goldberg, D. Grunwald, D. McCoy, S. Savage, and G. M. Voelker. “DefenestraTor: Throwing out windows in Tor”. In: *Privacy Enhancing Technologies*. Springer. 2011, pp. 134–154.

[5] M. AlSabah, K. Bauer, T. Elahi, and I. Goldberg. “The path less travelled: Overcoming tor’s bottlenecks with traffic splitting”. In: *Privacy Enhancing Technologies*. Springer. 2013, pp. 143–163.

[6] E. Andrés Miguel, C. Palamidessi, A. Sokolova, and P. Van Rossum. “Information Hiding in Probabilistic Concurrent Systems”. In: *Journal of Theoretical Computer Science (TCS)* 412.28 (2011), pp. 3072–3089.

[7] M. Backes and C. Jacobi. “Cryptographically Sound and Machine-Assisted Verification of Security Protocols”. In: *Proceedings of 20th International Symposium on Theoretical Aspects of Computer Science (STACS)*. 2003, pp. 675–686.

[8] M. Backes, A. Kate, and E. Mohammadi. “Ace: an efficient key-exchange protocol for onion routing”. In: *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*. ACM. 2012, pp. 55–64.

[9] M. Backes and B. Koepf. “Quantifying information flow in cryptographic systems”. In: *Mathematical Structures in Computer Science* 25.02 (2015), pp. 457–479.

[10] M. Backes and S. Meiser. “Differentially private smart metering with battery recharging”. In: *Data Privacy Management and Autonomous Spontaneous Security*. Springer, 2014, pp. 194–212.

[11] M. Backes, S. Meiser, and D. Schröder. “Delegatable Functional Signatures”. In: *Public-Key Cryptography–PKC 2016*. Springer, 2016, (to appear).

[12] M. Backes, S. Meiser, and M. Slowik. “Your Choice MATor (s)”. In: *Proceedings on Privacy Enhancing Technologies* 2016.2 (2015), pp. 40–60.

[13] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi. “AnoA: A Framework for Analyzing Anonymous Communication Protocols”. In: *Computer Security Foundations Symposium (CSF), 2013 IEEE 26th*. IEEE. 2013, pp. 163–178.

[14] M. Backes, A. Kate, S. Meiser, and E. Mohammadi. “(Nothing else) MATor (s): Monitoring the Anonymity of Tor’s Path Selection”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2014, pp. 513–524.

[15] M. Backes, I. Goldberg, A. Kate, and E. Mohammadi. “Provably Secure and Practical Onion Routing”. In: *Proc. 26st IEEE Symposium on Computer Security Foundations (CSF)*. 2012, pp. 369–385.

[16] M. Backes, A. Kate, S. Meiser, and T. Ruffing. “Secrecy without Perfect Randomness: Cryptography with (Bounded) Weak Sources.” In: *Proceedings of the 13th International Conference on Applied Cryptography and Network Security, (ACNS’15)*. 2015.

[17] M. V. Barbera, V. P. Kemerlis, V. Pappas, and A. D. Keromytis. “Cellflood: Attacking tor onion routers on the cheap”. In: *Computer Security–ESORICS 2013*. Springer, 2013, pp. 664–681.

[18] K. S. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. C. Sicker. “Low-resource routing attacks against tor”. In: *Proc. 6th ACM Workshop on Privacy in the Electronic Society (WPES)*. 2007, pp. 11–20.

[19] K. S. Bauer, M. Sherr, and D. Grunwald. “ExperimenTor: A Testbed for Safe and Realistic Tor Experimentation.” In: *Proceedings of the 4th Workshop on Cyber Security Experimentation and Test (CSET ’11)*. 2011.

[20] A. Beimel and S. Dolev. “Buses for Anonymous Message Delivery”. In: *Journal of Cryptology* 16.1 (Jan. 2003), pp. 25–39.

[21] M. Bhargava and C. Palamidessi. “Probabilistic Anonymity”. In: *CONCUR*. 2005, pp. 171–185.

[22] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz. “Denial of service or denial of security?” In: *Proceedings of the 14th ACM conference on Computer and communications security*. ACM. 2007, pp. 92–102.

[23] V. Boyko, P. D. MacKenzie, and S. Patel. "Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman". In: *Advances in Cryptology — EUROCRYPT*. 2000, pp. 156–171.

[24] C. Braun, K. Chatzikokolakis, and C. Palamidessi. "Quantitative notions of leakage for one-try attacks". In: *Electronic Notes in Theoretical Computer Science* 249 (2009), pp. 75–91.

[25] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson. "Touching from a Distance: Website Fingerprinting Attacks and Defenses". In: *Proc. 19th ACM Conference on Computer and Communication Security (CCS)*. 2012, pp. 605–616.

[26] J. Camenisch and A. Lysyanskaya. "A Formal Treatment of Onion Routing". In: *Advances in Cryptology — CRYPTO*. 2005, pp. 169–187.

[27] R. Canetti. "Universally Composable Security: A New Paradigm for Cryptographic Protocols". In: *Proc. 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*. 2001, pp. 136–145.

[28] R. N. C. Centre. *RIPE RIS Raw Data*. `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data`. accessed 2015.

[29] A. Chaabane, P. Manils, and M. A. Kaafar. "Digging into anonymous traffic: A deep analysis of the tor anonymizing network". In: *Network and System Security (NSS), 2010 4th International Conference on*. IEEE. 2010, pp. 167–174.

[30] S. Chakravarty, A. Stavrou, and A. D. Keromytis. "Traffic Analysis against Low-Latency Anonymity Networks Using Available Bandwidth Estimation". In: *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS)*. 2010, pp. 249–267.

[31] 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: *Passive and Active Measurement*. Springer. 2014, pp. 247–257.

[32] K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. "Anonymity protocols as noisy channels". In: *Trustworthy Global Computing*. Springer, 2007, pp. 281–300.

[33] D. Chaum. "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability". In: *J. Cryptology* 1.1 (1988), pp. 65–75.

[34] D. Chaum. "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". In: *Communications of the ACM* 4.2 (1981), pp. 84–88.

[35] C. Chen, D. E. Asoni, D. Barrera, G. Danezis, and A. Perrig. "HORNET: High-speed Onion Routing at the Network Layer". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security.* ACM. 2015, pp. 1441–1454.

[36] H. Corrigan-Gibbs and B. Ford. "Dissent: accountable anonymous group messaging". In: *Proceedings of the 17th ACM conference on Computer and communications security.* ACM. 2010, pp. 340–350.

[37] G. Danezis, C. Diaz, C. Troncoso, and B. Laurie. "\ text {Drac}: An Architecture for Anonymous Low-Volume Communications". In: *Privacy Enhancing Technologies.* Springer. 2010, pp. 202–219.

[38] C. Díaz. "Anonymity Metrics Revisited". In: *Anonymous Communication and its Applications.* 2006.

[39] C. Díaz, S. Seys, J. Claessens, and B. Preneel. "Towards Measuring Anonymity". In: *Proc. 2nd Workshop on Privacy Enhancing Technologies (PET).* 2002, pp. 54–68.

[40] W. Diffie, P. C. van Oorschot, and M. J. Wiener. "Authentication and Authenticated Key Exchanges". In: *Des. Codes Cryptography* 2.2 (1992), pp. 107–125.

[41] R. Dingledine and N. Mathewson. *Tor Path Specification.* `https://gitweb.torproject.org/torspec.git?a=blob_plain;hb=HEAD;f=path-spec.txt`. Accessed 2015.

[42] R. Dingledine, N. Mathewson, and P. Syverson. "Tor: The Second-Generation Onion Router". In: *Proc. 13th USENIX Security Symposium (USENIX).* 2004, pp. 303–320.

[43] C. Dwork. "Differential Privacy: A Survey of Results". In: *TAMC.* 2008, pp. 1–19.

[44] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. "Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail". In: *Security and Privacy (SP), 2012 IEEE Symposium on.* IEEE. 2012, pp. 332–346.

[45] M. Edman and P. Syverson. "AS-awareness in Tor path selection". In: *Proceedings of the 16th ACM conference on Computer and communications security.* ACM. 2009, pp. 380–389.

[46] J. Feigenbaum, A. Johnson, and P. F. Syverson. "A Model of Onion Routing with Provable Anonymity". In: *Proc. 11th Conference on Financial Cryptography and Data Security (FC).* 2007, pp. 57–71.

[47] J. Feigenbaum, A. Johnson, and P. F. Syverson. "Probabilistic Analysis of Onion Routing in a Black-Box Model". In: *Proc. 6th ACM Workshop on Privacy in the Electronic Society (WPES).* 2007, pp. 1–10.

[48] J. Feigenbaum, A. Johnson, and P. F. Syverson. "Probabilistic Analysis of Onion Routing in a Black-Box Model". In: *ACM Transactions on Information and System Security (TISSEC)* 15.3 (2012), p. 14.

[49] M. J. Freedman and R. Morris. "Tarzan: A Peer-to-Peer Anonymizing Network Layer". In: *Proc. 9th ACM Conference on Computer and Communications Security (CCS).* 2002, pp. 193–206.

[50] N. Gelernter and A. Herzberg. "On the limits of provable anonymity". In: *Proc. 12th ACM Workshop on Privacy in the Electronic Society (WPES).* 2013, pp. 225–236.

[51] B. Gierlichs, C. Troncoso, C. Díaz, B. Preneel, and I. Verbauwhede. "Revisiting a Combinatorial Approach toward Measuring Anonymity". In: *Proc. 7th ACM Workshop on Privacy in the Electronic Society (WPES).* 2008, pp. 111–116.

[52] Y. Gilad and A. Herzberg. "Spying in the dark: TCP and tor traffic analysis". In: *Privacy Enhancing Technologies.* Springer. 2012, pp. 100–119.

[53] S. Goel, M. Robson, M. Polte, and E. Sirer. *Herbivore: A scalable and efficient protocol for anonymous communication.* Tech. rep. Cornell University, 2003.

[54] O. Goldreich and Y. Lindell. "Session-Key Generation Using Human Passwords Only". In: *Advances in Cryptology — CRYPTO.* 2001, pp. 408–432.

[55] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. "Hiding routing information". In: *Information Hiding.* Springer. 1996, pp. 137–150.

[56] X. Gong, N. Borisov, N. Kiyavash, and N. Schear. "Website detection using remote traffic analysis". In: *Privacy Enhancing Technologies.* Springer. 2012, pp. 58–78.

[57] D. Gopal and N. Heninger. "Torchestra: Reducing interactive traffic delays over Tor". In: *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society.* ACM. 2012, pp. 31–42.

[58] J. Y. Halpern and K. R. O'Neill. "Anonymity and Information Hiding in Multiagent Systems". In: *Journal of Computer Security* 13.3 (2005), pp. 483–512.

[59] A. Hamel, J.-C. Grégoire, and I. Goldberg. *The Misentropists: New Approaches to Measures in Tor.* Tech. rep. Technical Report 2011-18, Cheriton School of Computer Science, University of Waterloo, 2011.

[60] J. Hayes and G. Danezis. "Guard Sets for Onion Routing". In: *Proceedings on Privacy Enhancing Technologies* 1 (2015).

[61] A. Hevia and D. Micciancio. "An Indistinguishability-Based Characterization of Anonymous Channels". In: *Proc. 8th Privacy Enhancing Technologies Symposium (PETS).* 2008, pp. 24–43.

[62] D. Hofheinz. "Possibility and Impossibility Results for Selective Decommitments". In: *J. Cryptology* 24.3 (2011), pp. 470–516.

[63] D. Hughes and V. Shmatikov. “Information Hiding, Anonymity and Privacy: a Modular Approach”. In: *Journal of Computer Security* 12.1 (2004), pp. 3–36.

[64] A. D. Jaggard, A. Johnson, S. Cortes, P. Syverson, and J. Feigenbaum. “20,000 in league under the sea: Anonymous communication, trust, MLATs, and undersea cables”. In: *Proceedings on Privacy Enhancing Technologies* 1.1 (2015), pp. 4–24.

[65] R. Jansen and N. Hooper. *Shadow: Running Tor in a box for accurate and efficient experimentation.* Tech. rep. DTIC Document, 2011.

[66] R. Jansen, N. Hopper, and Y. Kim. “Recruiting new Tor relays with BRAIDS”. In: *Proceedings of the 17th ACM conference on Computer and communications security.* ACM. 2010, pp. 319–328.

[67] R. Jansen, K. S. Bauer, N. Hopper, and R. Dingledine. “Methodically Modeling the Tor Network.” In: *Proceedings of the 5th Workshop on Cyber Security Experimentation and Test (CSET ’12).* 2012.

[68] R. Jansen, F. Tschorsch, A. Johnson, and B. Scheuermann. *The sniper attack: Anonymously deanonymizing and disabling the Tor network.* Tech. rep. DTIC Document, 2014.

[69] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson. “Users get routed: Traffic correlation on Tor by realistic adversaries”. In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security.* ACM. 2013, pp. 337–348.

[70] A. M. Johnson, P. Syverson, R. Dingledine, and N. Mathewson. “Trust-based anonymous communication: Adversary models and routing algorithms”. In: *Proceedings of the 18th ACM conference on Computer and communications security.* ACM. 2011, pp. 175–186.

[71] J. Juen, A. Das, A. Johnson, N. Borisov, and M. Caesar. “Defending tor from network adversaries: A case study of network path prediction”. In: *Proceedings on Privacy Enhancing Technologies* 1 (2015).

[72] A. Kate, G. M. Zaverucha, and I. Goldberg. “Pairing-based onion routing with improved forward secrecy”. In: *ACM Transactions on Information and System Security (TISSEC)* 13.4 (2010), p. 29.

[73] R. Kusters, T. Truderung, and A. Vogt. “Formal analysis of chaumian mix nets with randomized partial checking”. In: *Security and Privacy (SP), 2014 IEEE Symposium on.* IEEE. 2014, pp. 343–358.

[74] V. Liu, S. Han, A. Krishnamurthy, and T. Anderson. “Tor instead of IP”. In: *Proceedings of the 10th ACM Workshop on Hot Topics in Networks.* ACM. 2011, p. 14.

[75] K. Loesing, S. J. Murdoch, and R. Dingledine. “A case study on measuring statistical data in the Tor anonymity network”. In: *Financial Cryptography and Data Security.* Springer, 2010, pp. 203–215.

[76] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. “iPlane: An information plane for distributed services”. In: *Proceedings of the 7th symposium on Operating systems design and implementation.* USENIX Association. 2006, pp. 367–380.

[77] N. Mathewson and R. Dingledine. “Practical traffic analysis: Extending and resisting statistical disclosure”. In: *Privacy Enhancing Technologies.* Springer. 2005, pp. 17–34.

[78] S. Mauw, J. H. Verschuren, and E. P. de Vink. “A Formalization of Anonymity and Onion Routing”. In: *Proc. 9th European Symposium on Research in Computer Security (ESORICS).* 2004, pp. 109–124.

[79] *Maxind GeoIP2-City Database.* `https://www.maxmind.com/de/geoip2-databases`. accessed 2015.

[80] J. McLachlan, A. Tran, N. Hopper, and Y. Kim. “Scalable onion routing with torsk”. In: *Proceedings of the 16th ACM conference on Computer and communications security.* ACM. 2009, pp. 590–599.

[81] I. Mironov, O. Pandey, O. Reingold, and S. P. Vadhan. “Computational Differential Privacy”. In: *Advances in Cryptology — CRYPTO.* Vol. 5677. 2009, pp. 126–142.

[82] P. Mittal, F. G. Olumofin, C. Troncoso, N. Borisov, and I. Goldberg. “PIR-Tor: Scalable Anonymous Communication Using Private Information Retrieval.” In: *USENIX Security Symposium.* 2011.

[83] P. Mittal, A. Khurshid, J. Juen, M. Caesar, and N. Borisov. “Stealthy traffic analysis of low-latency anonymous communication using throughput fingerprinting”. In: *Proceedings of the 18th ACM conference on Computer and communications security.* ACM. 2011, pp. 215–226.

[84] W. B. Moore, C. Wacek, and M. Sherr. “Exploring the potential benefits of expanded rate limiting in tor: Slow and steady wins the race with tortoise”. In: *Proceedings of the 27th Annual Computer Security Applications Conference.* ACM. 2011, pp. 207–216.

[85] S. J. Murdoch and R. N. M. Watson. “Metrics for Security and Performance in Low-Latency Anonymity Systems”. In: *Proc. 8th Privacy Enhancing Technologies Symposium (PETS).* 2008, pp. 115–132.

[86] S. J. Murdoch and G. Danezis. “Low-cost traffic analysis of Tor”. In: *Security and Privacy, 2005 IEEE Symposium on.* IEEE. 2005, pp. 183–195.

[87] S. J. Murdoch and P. Zieliński. “Sampled traffic analysis by internet-exchange-level adversaries”. In: *Privacy Enhancing Technologies.* Springer. 2007, pp. 167–183.

[88] A. Nambiar and M. Wright. “Salsa: a structured approach to large-scale anonymity”. In: *Proceedings of the 13th ACM conference on Computer and communications security.* ACM. 2006, pp. 17–26.

[89] A. Neil. *SelekTOR - Tor Exit Node Selection Made Simple.* Accessed February, 2015.

[90] *Ookla's NetIndex.* Accessed July, 2015.

[91] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel. “Website fingerprinting in onion routing based anonymization networks”. In: *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society.* ACM. 2011, pp. 103–114.

[92] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel. “Website Fingerprinting in Onion Routing Based Anonymization Networks”. In: *Proc. 10th ACM Workshop on Privacy in the Electronic Society (WPES).* 2011, pp. 103–114.

[93] A. Pfitzmann and M. Hansen. *A Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management.* http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf. v0.34. Aug. 2010.

[94] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. “Anonymous connections and onion routing”. In: *Selected Areas in Communications, IEEE Journal on* 16.4 (1998), pp. 482–494.

[95] M. K. Reiter and A. D. Rubin. “Crowds: Anonymity for Web Transactions”. In: *ACM Transactions on Information and System Security (TISSEC)* 1.1 (1998), pp. 66–92.

[96] *Routeviews mapping, IP to AS.* Accessed October, 2015.

[97] M. Schuchard, A. W. Dean, V. Heorhiadi, N. Hopper, and Y. Kim. “Balancing the shadows”. In: *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society.* ACM. 2010, pp. 1–10.

[98] A. Serjantov and G. Danezis. “Towards an Information Theoretic Metric for Anonymity”. In: *Proc. 2nd Workshop on Privacy Enhancing Technologies (PET).* 2002, pp. 41–53.

[99] F. Shirazi, M. Goehring, and C. Diaz. “Tor experimentation tools”. In: *Security and Privacy Workshops (SPW), 2015 IEEE.* IEEE. 2015, pp. 206–213.

[100] V. Shmatikov. “Probabilistic Analysis of an Anonymity System”. In: *Journal of Computer Security* 12.3-4 (2004), pp. 355–377.

[101] V. Shmatikov and M.-H. Wang. “Measuring Relationship Anonymity in Mix Networks”. In: *Proc. 7th ACM Workshop on Privacy in the Electronic Society (WPES).* 2006, pp. 59–62.

[102] V. Shmatikov and M.-H. Wang. “Timing analysis in low-latency mix networks: Attacks and defenses”. In: *Computer Security–ESORICS 2006.* Springer, 2006, pp. 18–33.

[103] *Source-code of MATor.* available at `https://www.infsec.cs.uni-saarland.de/projects/anonymity-guarantees/mator.html`.

[104] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal. “RAPTOR: Routing Attacks on Privacy in Tor”. In: *arXiv preprint arXiv:1503.03940* (2015).

[105] E. Syta, H. Corrigan-Gibbs, S.-C. Weng, D. Wolinsky, B. Ford, and A. Johnson. “Security analysis of accountable anonymity in Dissent”. In: *ACM Transactions on Information and System Security (TISSEC)* 17.1 (2014), p. 4.

[106] P. Syverson. “Why I’m not an entropist”. In: *Security Protocols XVII.* Springer, 2013, pp. 213–230.

[107] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr. “Towards an Analysis of Onion Routing Security”. In: *Proc. Workshop on Design Issues in Anonymity and Unobservability (WDIAU).* 2000, pp. 96–114.

[108] C. Tang and I. Goldberg. “An improved algorithm for Tor circuit scheduling”. In: *Proceedings of the 17th ACM conference on Computer and communications security.* ACM. 2010, pp. 329–339.

[109] *The Tor Project.* `https://www.torproject.org/`. Accessed Feb 2013. 2003.

[110] *Tor Metrics Portal.* `https://metrics.torproject.org/`. Accessed Feb 2013.

[111] G. Tóth, Z. Hornák, and F. Vajda. “Measuring anonymity revisited”. In: *Proceedings of the Ninth Nordic Workshop on Secure IT Systems.* Espoo, Finland. 2004, pp. 85–90.

[112] *University of Oregon Route Views Project.* `http://www.routeviews.org/`. accessed 2015.

[113] C. Wacek, H. Tan, K. S. Bauer, and M. Sherr. “An Empirical Evaluation of Relay Selection in Tor.” In: *Proc. 20th Annual Network & Distributed System Security Symposium (NDSS).* 2013.

[114] T. Wang and I. Goldberg. “Improved website fingerprinting on tor”. In: *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society.* ACM. 2013, pp. 201–212.

[115] T. Wang, K. Bauer, C. Forero, and I. Goldberg. “Congestion-aware path selection for Tor”. In: *Financial Cryptography and Data Security.* Springer, 2012, pp. 98–113.

[116] P. Winter, R. Köwer, M. Mulazzani, M. Huber, S. Schrittwieser, S. Lindskog, and E. Weippl. “Spoiled onions: Exposing malicious Tor exit relays”. In: *Privacy Enhancing Technologies*. Springer. 2014, pp. 304–331.

[117] J. Zhang, C. Chen, Y. Xiang, and W. Zhou. “Robust network traffic identification with unknown applications”. In: *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*. ACM. 2013, pp. 405–414.

Appendix A

Appendix

A.1 Abstracting Tor in UC

We cite the the description of the ideal functionality $\mathcal{F}_{\mathrm{OR}}$. Sections A.1.1 and A.1.2 are taken from [15].

A.1.1 System and Adversary Model

We consider a fully connected network of $n + m$ parties $\mathbb{N} = \{P_1, \ldots, P_n, \ldots, P_{n+1}, \ldots, P_{n+m}\}$. We consider the parties $P_1, \ldots, P_n$ to be OR nodes, and the parties $P_{n+1}, \ldots, P_{n+m}$ to only be senders. We furthermore assume that the set of OR nodes is publicly known. The onion routers can be compromised by the adversary by sending `compromise` messages. The senders, however, can in our model not be compromised, since the adversary can just act as a sender of the OR network. Formally, $P_{n+1}, \ldots, P_{n+m}$, consequently, do not react towards `compromise` messages.

Tor has not been designed to resist against global adversaries. Such an adversary is too strong for many practical purposes as it can simply break the anonymity of an OR protocol by holding back all but one onion and tracing that one onion though the network. However, in contrast to previous work, we do not only consider local adversaries, which do not control more than the compromised OR routers, but also partially global adversaries that control a certain portion of the network. Analogous to the network functionality $\mathcal{F}_{\mathrm{SYN}}$ proposed by Canetti [27], we model the network as an ideal functionality $\mathcal{F}_{\mathrm{NET}}$, which bounds the number of adversary-controlled links to $q \in [0, \binom{n}{2}]$. For adversary-controlled links the messages are forwarded to the adversary; otherwise, they are directly delivered.

Let $\mathcal{R}$ represent all possible recipients $\{S_1, \ldots, S_\Delta\}$ which reside in the network abstracted by a network functionality $\mathcal{F}_{\mathrm{NET}}q$.

We stress that the UC framework does not provide a notion of time; hence, the analysis of timing attacks, such as traffic analysis, is not in the scope of this work.

Adaptive Corruptions Forward secrecy [40] is an important property for onion routing. In order to analyze this property, we allow adaptive corruptions of nodes by the adversary A. Such an adaptive corruption is formalized by a message `compromise` , which is sent to the respective party. Upon such a `compromise` message the internal state of that party is deleted and a long-term secret key `sk` for the node is revealed to the adversary. A can then impersonate the node in the future; however, A cannot obtain the information about its ongoing sessions. We note that this restriction arises due to the currently available security proof techniques and the well-known selective opening problem with symmetric encryptions [62], and the restriction is not specific to our constructions [54]. We could also restrict ourselves to a static adversary as in previous work [26]; however, that would make an analysis of forward secrecy impossible.

A.1.2 Ideal Functionality

The presentation of the ideal functionality $\mathcal{F}_{\mathrm{OR}}$ is along the lines of the description OR protocol Π_{OR} from Section [15, Section 2.4]. We continue to use the message-based state transitions from Π_{OR}, and consider sub-machines for all n nodes in the ideal functionality. To communicate with each other through messages and data structures, these sub-machines share a memory space in the functionality. The sub-machine pseudocode for the ideal functionality appears in Figure A.1 and three subroutines are defined in Figure A.3. As the similarity between pseudocodes for the OR protocol and the ideal functionality is obvious, rather than explaining the OR message flows again, we concentrate on the differences.

The only major difference between Π_{OR} and $\mathcal{F}_{\mathrm{OR}}$ is that cryptographic primitives such as message wrapping, unwrapping, and key exchange are absent in the ideal world; we do not have any keys in $\mathcal{F}_{\mathrm{OR}}$, and the OR messages `WrapOnion` and `UnwrapOnion` as well as the 1W-AKE messages INITIATE, RESPOND, and COMPUTEKEY are absent.

The ideal functionality also abstracts the directory server and expects on the input/output interface of $\mathcal{F}_{\mathrm{REG}}$ (from the setting with Π_{OR}) an initial message with the list $\langle P_i \rangle_{i=1}^{n}$ of valid nodes. This initial message corresponds to the list of onion routers that have been approved by an administrator. We call the part of $\mathcal{F}_{\mathrm{OR}}$ that abstracts the directory servers DIR. For the sake of brevity, we do not present the pseudocode of DIR . Upon an initial message with a list $\langle P_i \rangle_{i=1}^{n}$ of valid nodes, DIR waits for all nodes P_i ($i \in \{1, \dots, n\}$) for a message (`register`, P_i). Once all nodes registered, DIR sends a message (`registered`, $\langle P_i \rangle_{i=1}^{n}$) with a list of valid and registered nodes to every party that registered, and to every party that sends a `last` message to DIR.

A.1.3 Ideal Functionality for a Party P

We assume that the Tor protocol manages a list of self-generated party identifiers (for all Tor entities), to specify the recipient of any message. Upon an input (`create circuit`, $\langle P, P_1, , \dots, P_l \rangle$), P creates a fresh circuit C, initially only consisting of P, and then iteratively extends the circuit via the subroutine ExtendCircuit, as shown in Cref-figure:SubFOR. To this end, it sends messages `create` to all involved parties, however using the already established connections, where each subsequent extension is triggered by a message `extended`, specifying that the previous extension was successful. Upon an input $(\texttt{send}, \mathsf{C}, m)$, P checks whether the circuit can still be used for transmitting another message (their number is bound by a limit `ttl-count`), and if so, sends the message m together with the circuit ID to the guard node that will relay it to further Tor nodes until it reaches the recipient. Otherwise, it destroys the circuit via the subrouting DestroyCircuit (c.f. Figure A.3).

Messages from A **and** $\mathcal{F}_{\text{NET}}$ In Figure A.1 and Figure A.4, we present the pseudocode for the adversary messages and the network functionality, respectively. For our basic analysis, we model an adversary that can control all communication links and servers in $\mathcal{F}_{\text{NET}}$, but cannot view or modify messages between parties due to the presence of the secure and authenticated channel. Therefore, sub-machines in the functionality store their messages in the shared memory, and create and send handles $\langle P, P_{\texttt{next}}, h\rangle$ for these messages in $\mathcal{F}_{\text{NET}}$. The message length does not need to be leaked as we assume a fixed message size (for all $M(\kappa)$). Here, P is the sender, $P_{\texttt{next}}$ is the receiver and h is a handle or a pointer to the message in the shared memory of the ideal functionality. In our analysis, all $\mathcal{F}_{\text{NET}}$ messages flow to A, which may choose to return these handles back to $\mathcal{F}_{\text{OR}}$ through $\mathcal{F}_{\text{NET}}$ at its own discretion. However, $\mathcal{F}_{\text{NET}}$ also maintains a mechanism through OBSERVED flags for the non-global adversary A. The adversary may also corrupt or replay the corresponding messages; however, these active attacks are always detected by the receiver due to the presence of a secure and authenticated channel between any two communicating parties and we need not model these corruptions.

The adversary can compromise a party P or server S by sending a `compromise` message to respectively $\mathcal{F}_{\text{OR}}$ and $\mathcal{F}_{\text{NET}}$. For party P or server S, the respective functionality then sets the `compromised` tag to `true`. Furthermore, all input or network messages that are supposed to be visible to the compromised entity are forwarded to the adversary. In principle, the adversary runs that entity for the rest of the protocol and can send messages from that entity. In that case, it can also propagate corrupted messages which in Π_{OR} can only be detected during `UnwrapOnion` calls at OP or the exit node. We model these corruptions using $\texttt{corrupted}(msg) = \{\texttt{true}, \texttt{false}\}$ status flags, where $\texttt{corrupted}(msg)$ status of messages is maintained across nodes until they reach end nodes. Furthermore, for every corrupted message, the adversary also provides a modification function $T(\cdot)$ as the end nodes run by the adversary may continue execution even after observing a `corrupted` flag. In that case, $T(\cdot)$ captures the exact modification made by the adversary.

We stress that $\mathcal{F}_{\text{OR}}$ does not need to reflect reroutings and circuit establishments initiated by the adversary, because the adversary learns, loosely speaking, no new information by rerouting onions.[1] Similar to the previous work [26], a message is directly given to the adversary if all remaining nodes in a communication path are under adversary control.

[1]More formally, the simulator can compute all responses for rerouting or such circuit establishments without requesting information from $\mathcal{F}_{\mathrm{OR}}$ because the simulator knows all long-term and session keys. The only information that the simulator does not have is the routing information, which the simulator gets in case of rerouting or circuit establishment.

```
upon input $(\texttt{create circuit}, \textsc{Parties} = \langle P, P_1, \ldots, P_\ell \rangle)$:
  store Parties and $\mathsf{C}$:=$\langle P \rangle$
  Run $\textsc{ExtendCircuit}(\textsc{Parties}, \mathsf{C})$
upon input $(\texttt{send}, \mathsf{C} = \langle P \overset{\text{CID}_1}{\Leftrightarrow} P_1 \Leftrightarrow \cdots P_\ell \rangle, m)$:
  if $\mathit{Used}(\text{CID}_1) <$ ttl-count then
    $\mathit{Used}(\text{CID}_1)$++; $\textsc{SendMessage}(P_1, \text{CID}_1, \texttt{relay}, \langle \texttt{data}, m \rangle)$
  else
    $\textsc{DestroyCircuit}(\mathsf{C}, \text{CID}_1)$; output $(\texttt{destroyed}, \mathsf{C}, m)$
  end if
upon receiving a handle $\langle P, P_{\texttt{next}}, h \rangle$ from $\mathcal{F}_{\text{NET}}$:
  send $(msg)$:=lookup$(h)$ to a receiving submachine $P_{\texttt{next}}$
upon receiving a msg $(P_i, \text{CID}, \texttt{create})$ through a handle:
  store $\mathsf{C}$:=$\langle P_i \overset{\text{CID}}{\Leftrightarrow} P \rangle$; $\textsc{SendMessage}(P_i, \text{CID}, \texttt{created})$
upon receiving a msg $(P_i, \text{CID}, \texttt{created})$ through a handle:
  if $\texttt{prev}(\text{CID}) = (P', \text{CID}')$ then
    $\textsc{SendMessage}(P', \text{CID}', \texttt{relay}, \texttt{extended})$
  else if $\texttt{prev}(\text{CID}) = \perp$ then
    $\textsc{ExtendCircuit}(\textsc{Parties}, \mathsf{C})$
  end if
upon receiving a msg $(P_i, \text{CID}, \texttt{relay}, O)$ through a handle:
  Run $\textsc{RelayMessage}(P_i, \text{CID}, O)$
upon receiving a msg $(\text{SID}, m)$ from $\mathcal{F}_{\text{NET}}$:
  obtain $\mathsf{C} = \langle P' \overset{\text{CID}}{\Leftrightarrow} P \rangle$ for SID
  $\textsc{SendMessage}(P', \text{CID}, \texttt{relay}, \langle \texttt{data}, m \rangle)$
upon receiving a msg $(P_i, \text{CID}, \texttt{destroy})$ through a handle:
  $\textsc{DestroyCircuit}(\mathsf{C}, \text{CID})$
receiving a msg $(P_i, P, [\texttt{cmd}, ]h, [\texttt{corrupt}, T(\cdot)])$from A:
  $(\mathit{message})$:=lookup$(h)$
  if corrupt $=$ true then
    $\mathit{message}$:=$T(msg)$; set corrupted$(\mathit{message})$:=true
  end if
  process $\mathit{message}$ as if the receiving submachine was $P$
upon receiving a msg $(\texttt{compromise}, P)$ from A:
  set compromised$(P)$:=true
  delete all local information at $P$
```

Figure A.1: The ideal functionality $\mathcal{F}_{\text{OR}}$ for Party P [15, Fig.5]

```
$\textsc{RelayMessage}(P_i, \textsc{cid}, \texttt{relay}, O)$:
  if $\texttt{prev}(\textsc{cid}) = \perp$ then
    if $\texttt{next}(\textsc{cid}) = \perp$ then
      get $(\texttt{type}, m)$ from $O$
    end if
    else $(P', \textsc{cid}') \texttt{:=} \texttt{next}(\textsc{cid})$
  else
    $(P', \textsc{cid}') \texttt{:=} \texttt{prev}(\textsc{cid})$
  end if
  switch $(\texttt{type})$
  case $\texttt{extend circuit}$ :
    get $P_{\texttt{next}}$ from $m$; $\textsc{cid}_{\texttt{next}} \leftarrow \{0,1\}^{\kappa}$
    update $\mathsf{C} \texttt{:=} \langle P_i \overset{\textsc{cid}}{\Leftrightarrow} P \overset{\textsc{cid}_{\texttt{next}}}{\Leftrightarrow} P_{\texttt{next}} \rangle$
    $\textsc{SendMessage}(P_{\texttt{next}}, \textsc{cid}_{\texttt{next}}, \texttt{create})$
  case $\texttt{extended}$ :
    update $\mathsf{C}$ with $P_{\texttt{added}}$; $\textsc{ExtendCircuit}(\textsc{Parties}, \mathsf{C})$
  case $\texttt{data}$ :
    if $(\textsc{cid} \in \textsc{MyCircuits})$ then output $(\texttt{received}, C, m)$
    else if $m = (S, m')$
      generate or lookup the unique $\textsc{sid}$ for $\textsc{cid}$
      send $(P, S, \textsc{sid}, m')$ to $\mathcal{F}_{\textsc{Net}}q$
  case $\texttt{corrupted}$ : /*corrupted onion*/
    $\textsc{DestroyCircuit}(\mathsf{C}, \textsc{cid})$
  case $\texttt{default}$ : /*encrypted forward/backward onion*/
    $\textsc{SendMessage}(P', \textsc{cid}', \texttt{relay}, O)$
```

Figure A.2: The RELAYMESSAGE subroutine of the ideal functionality $\mathcal{F}_{\text{OR}}$ for Party P [15, Fig.5]

```
$\textsc{ExtendCircuit}(\textsc{Parties} = (P_j)_{j=1}^{\mathtt{p}}, \mathsf{C} = \langle P \overset{\textsc{cid}_1}{\Leftrightarrow} P_1 \Leftrightarrow \cdots P_{\ell'}\rangle)$:
  determine the next node $P_{\ell'+1}$ from $\textsc{Parties}$ and $\mathsf{C}$
  if $P_{\ell'+1} = \perp$ then
    output $(\mathtt{created}, \mathsf{C})$
  else
    if $P_{\ell'+1} = P_1$ then
      $\textsc{cid}_1 \leftarrow \{0,1\}^\kappa$; $\textsc{SendMessage}(P_1, \textsc{cid}_1, \mathtt{create})$
      Store $\textsc{cid}_1$ in $\textsc{MyCircuits}$
    else
      $\textsc{SendMessage}(P_1, \textsc{cid}_1, \mathtt{relay}, \{\mathtt{extend\ circuit}, P_{\ell'+1}\})$
    end if
  end if
$\textsc{DestroyCircuit}(\mathsf{C}, \textsc{cid})$:
  if $\mathtt{next}(\textsc{cid}) = (P_{\mathtt{next}}, \textsc{cid}_{\mathtt{next}})$ then
    $\textsc{SendMessage}(P_{\mathtt{next}}, \textsc{cid}_{\mathtt{next}}, \mathtt{destroy})$
  else if $\mathtt{prev}(\textsc{cid}) = (P_{\mathtt{prev}}, \textsc{cid}_{\mathtt{prev}})$ then
    $\textsc{SendMessage}(P_{\mathtt{prev}}, \textsc{cid}_{\mathtt{prev}}, \mathtt{destroy})$
  end if
  discard $\mathsf{C}$ and all streams
$\textsc{SendMessage}(P_{\mathtt{next}}, \textsc{cid}_{\mathtt{next}}, \mathtt{cmd}, [\mathtt{relay\text{-}type}], [\mathtt{data}])$:
  create a $msg$ for $P_{\mathtt{next}}$ from the input
  draw a fresh handle $h$ and set $\mathtt{lookup}(h)\mathtt{:=}msg$
  if $\mathtt{compromised}(P_{\mathtt{next}}) = \mathtt{true}$ then
    let $P_{\mathtt{last}}$ be the last node in the contiguous $\mathtt{compromised}$ path starting in $P_{\mathtt{next}}$.
    To this end, reconstruct the remaining circuit $P_{\mathtt{next}}, \ldots, P_{\mathtt{last}}$ from the shared
    memory by iteratively using $\mathtt{next}$, starting with $\textsc{cid}_{\mathtt{next}}$.
    if $(P_{\mathtt{last}} = OP)$ or $P_{\mathtt{last}}$ is the exit node then
      send $\langle P, P_{\mathtt{next}}, \ldots, P_{\mathtt{last}}, \textsc{cid}_{\mathtt{next}}, \mathtt{cmd}, msg\rangle$ to A
    else
      send $\langle P, P_{\mathtt{next}}, \ldots, P_{\mathtt{last}}, \textsc{cid}_{\mathtt{next}}, \mathtt{cmd}, h\rangle$ to A
    end if
  else
    send $\langle P, P_{\mathtt{next}}, h\rangle$ to $\mathcal{F}_{\textsc{Net}}q$
  end if
```

Figure A.3: Subroutines of $\mathcal{F}_{\mathrm{OR}}$ for Party P[15, Fig.6]

```
$\mathcal{F}_{\text{NET}}$ receiving a msg $(\texttt{observe}, P, P_{\texttt{next}})$ from A:
  set $\textsc{observed}(P, P_{\texttt{next}})\texttt{:=true}$
**upon** receiving a msg $(\texttt{compromise}, S)$ from A:
  set $\texttt{compromised}(S)\texttt{:=true}$; send A all existing $\textsc{sid}$ of $S$ (as stored below).
**upon** receiving a msg $(P, P_{\texttt{next}}/S, m)$ from $\mathcal{F}_{\text{OR}}$:
  **if** $P_{\texttt{next}}/S$ is a $\mathcal{F}_{\text{OR}}$ node **then**
    **if** $\textsc{observed}(P, P_{\texttt{next}}) = \texttt{true}$ **then**
      forward the msg $(P, P_{\texttt{next}}, m)$ to A
    **else**
      reflect the msg $(P, P_{\texttt{next}}, m)$ to $\mathcal{F}_{\text{OR}}$
    **end if**
  **else if** $P_{\texttt{next}}/S$ is a $\mathcal{F}_{\text{NET}}$ server **then**
    **if** $m$ is of the form $(m', \textsc{sid})$ **then**
      Store $\textsc{sid}$ for $P_{\texttt{next}}/S$.
      **if** $\texttt{compromised}(S) = \texttt{true}$ or $\textsc{observed}(P, S) = \texttt{true}$ **then**
        forward the msg $(P, S, m)$ to A
      **else**
        output $(P, S, m')$
      **end if**
    **end if**
  **end if**
**upon** receiving a msg $(P/S, P_{\texttt{next}}, m)$ from A:
  forward the msg $(P/S, P_{\texttt{next}}, m)$ to $\mathcal{F}_{\text{OR}}$
```

Figure A.4: The Network Functionality $\mathcal{F}_{\text{NET}}$ [15, Fig.7]: A/B denotes that as a variable name either A or B is used.