2013 IEEE 26th Computer Security Foundations Symposium

AnoA: A Framework For Analyzing Anonymous Communication Protocols
Unified Definitions and Analyses of Anonymity Properties

Michael Backes
Saarland University and MPI-SWS
backes@cs.uni-saarland.de

Aniket Kate
MMCI, Saarland University
aniket@mmci.uni-saarland.de

Praveen Manoharan
Saarland University
manoharan@cs.uni-saarland.de

Sebastian Meiser
Saarland University
meiser@cs.uni-saarland.de

Esfandiar Mohammadi
Saarland University
mohammadi@cs.uni-saarland.de

***Abstract*—Protecting individuals' privacy in online communications has become a challenge of paramount importance. To this end, anonymous communication (AC) protocols such as the widely used Tor network have been designed to provide anonymity to their participating users. While AC protocols have been the subject of several security and anonymity analyses in the last years, there still does not exist a framework for analyzing complex systems such as Tor and their different anonymity properties in a unified manner.**

In this work we present AnoA: a generic framework for defining, analyzing, and quantifying anonymity properties for AC protocols. AnoA relies on a novel relaxation of the notion of (computational) differential privacy, and thereby enables a unified quantitative analysis of well-established anonymity properties, such as sender anonymity, sender unlinkability, and relationship anonymity. While an anonymity analysis in AnoA can be conducted in a purely information theoretical manner, we show that the protocol's anonymity properties established in AnoA carry over to secure cryptographic instantiations of the protocol. We exemplify the applicability of AnoA for analyzing real-life systems by conducting a thorough analysis of the anonymity properties provided by the Tor network against passive attackers. Our analysis significantly improves on known anonymity results from the literature.

***Keywords*-anonymity analysis; differential privacy; unlinkability; relationship anonymity; Tor**

I. Introduction

Protecting individuals' privacy in online communications has become a challenge of paramount importance. A wide variety of privacy enhancing technologies, comprising many different approaches, have been proposed to solve this problem. Privacy enhancing technologies, such as anonymous communication (AC) protocols, seek to protect users' privacy by anonymizing their communication over the Internet. Employing AC protocols has become increasingly popular over the last decade. This popularity is exemplified by the success of the Tor network [1].

There has been a substantial amount of previous work [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] on analyzing the anonymity provided by various AC protocols such as dining cryptographers network (DC-net) [15], Crowds [16], mix network (Mixnet) [17], and onion routing (e.g., Tor) [18]. However, most of the previous works only consider a single anonymity property for a particular AC protocol under a specific adversary scenario. Previous frameworks such as [19] only guarantee anonymity for a symbolic abstraction of the AC, not for its cryptographic realization. Moreover, while some existing works like [14] consider an adversary with access to *a priori* probabilities for the behavior of users, there is still no work that is capable of dealing with an adversary that has arbitrary auxiliary information about user behavior.

Prior to this work, there is no framework that is both expressive enough to unify and compare relevant anonymity notions (such as sender anonymity, sender unlinkability, and relationship anonymity), and that is also well suited for analyzing complex cryptographic protocols.

A. Contributions

In this work, we make three contributions to the field of anonymity analysis.

As a first contribution, we present the novel anonymity analysis framework AnoA. In AnoA we define and analyze anonymity properties of AC protocols. Our anonymity definition is based on a novel generalization of differential privacy, a notion for privacy preserving computation that has been introduced by Dwork et al. [20], [21]. The strength of differential privacy resides in a strong adversary that has maximal control over two adjacent settings that it has to distinguish. However, applying differential privacy to AC protocols seems impossible. While differential privacy does not allow for leakage of (potentially private) data, AC protocols inherently leak to the recipient the data that a sender sends to this

DOI 10.1109/CSF.2013.18

IEEE computer society

recipient. We overcome this contradiction by generalizing the adjacency of settings between which an adversary has to distinguish. We introduce an explicit *adjacency function α* that characterizes whether two settings are considered adjacent or not. In contrast to previous work on anonymity properties, this generalization of differential privacy, which we name α-IND-CDP, is based on IND-CDP [22] and allows the formulation of anonymity properties in which the adversary can choose the messages—which results in a strong adversary—as long as the adjacent challenge inputs carry the same messages. Moreover, AnoA is compatible with simulation-based composability frameworks, such as UC [23], IITM [24], or RSIM [25]. In particular, for all protocols that are securely abstracted by an ideal functionality [26], [27], [28], [29], [30], our definitions allow an analysis of these protocols in a purely information theoretical manner.

As a second contribution, we formalize the well-established notions of sender anonymity, (sender) unlinkability, and relationship anonymity in our framework, by introducing appropriate adjacency functions. We discuss why our anonymity definitions accurately capture these notions, and show for sender anonymity and (sender) unlinkability that our definition is equivalent to the definitions from the literature. For relationship anonymity, we argue that previous formalizations captured recipient anonymity rather than relationship anonymity, and we discuss the accuracy of our formalization. Moreover, we show relations between our formalizations of sender anonymity, (sender) unlinkability, and relationship anonymity: sender anonymity implies both (sender) unlinkability and relationship anonymity, but is not implied by either of them.

As a third contribution, we apply our framework to the most successful AC protocol—Tor. Since the underlying cryptographic model does not capture system-level attacks, we model known system-level attacks, such as website fingerprinting and traffic correlation, as an over-approximation of the ideal functionality. In addition, we discuss a known countermeasure for Tor's high sensitivity to compromised nodes: the entry guards mechanism. We show that using entry guards dramatically reduces the adversary's success probability and why this is the case. We leverage previous results that securely abstract Tor as an ideal functionality (in the UC framework) [30]. Then, we illustrate that proving sender anonymity, sender unlinkability, and relationship anonymity against passive adversaries boils down to a combinatoric analysis, purely based on the number of corrupted nodes in the network.

Outline of the Paper. In Section II we introduce the notation used throughout the paper. Section III presents our anonymity analysis framework AnoA and introduces the formalizations of sender anonymity, unlinkability, and relationship anonymity notions in the framework. Section IV compares our anonymity notions with those from the literature as well as with each other. In Section V, we demonstrate compatibility of AnoA with a simulation-based composability framework (in particular, the UC framework), and we apply the corresponding preservation result to analyze the Tor network in Section VI. Finally, we conclude and discuss some further interesting directions in Section VIII.

II. Notation

Before we present AnoA, we briefly introduce some of the notation used throughout the paper. We differentiate between two different kinds of assignments: $a := b$ denotes a being assigned the value b, and $a \leftarrow \beta$ denotes that a value is drawn from the distribution β and a is assigned the outcome. In a similar fashion $i \overset{R}{\leftarrow} I$ denotes that i is drawn uniformly at random from the set I.

Probabilities are given over a probability space which is explicitly stated unless it is clear from context. For example $\Pr[b = 1 : b \overset{R}{\leftarrow} \{0, 1\}]$ denotes the probability of the event $b = 1$ in the probability space where b is chosen uniformly at random from the set $\{0, 1\}$.

Our security notion is based on interacting Turing Machines (TM). We use an oracle-notation for describing the interaction between an adversary and a challenger: $\mathcal{A}^{\mathcal{B}}$ denotes the interaction of TM $\mathcal{A}$ with TM $\mathcal{B}$ where $\mathcal{A}$ has oracle access to $\mathcal{B}$. Whenever $\mathcal{A}$ activates $\mathcal{B}$ again, $\mathcal{B}$ will continue its computation on the new input, using its previously stored state. $\mathcal{A}$ can then again activate $\mathcal{B}$ with another input value, and $\mathcal{B}$ will continue its computation with the new input, using its previously stored state. This interaction continues until $\mathcal{A}$ returns an output, which is considered the output of $\mathcal{A}^{\mathcal{B}}$.

In this paper we focus on computational security, i.e. all machines are computationally bounded. More formally, we consider *probabilistic, polynomial time* (PPT) TMs, which we denote with PPT whenever required.

III. The AnoA Framework

In this section, we present the AnoA framework and our formulations of sender anonymity, sender unlinkability, and relationship anonymity (Section III-C). These formulations are based on a novel generalization of differential privacy that we describe in Section III-B. Before we introduce this notion, we first describe the underlying protocol model. Using our protocol model, AC protocols are closely related to mechanisms that process databases, a fact that enables us to apply a more flexible form of *differential privacy.*

A. Protocol model

Anonymous communication (AC) protocols are distributed protocols that enable multiple users to anonymously communicate with multiple recipients. Formally, an AC protocol is an interactive Turing machine.[1] We associate a protocol with a user space $\mathcal{U}$, a recipient space $\mathcal{R}$ and an auxiliary information space Aux. Users' actions are modeled as an input to the protocol and represented in the form of an ordered *input table*. Each row in the input table contains a user $u \in \mathcal{U}$ that performs some action, combined with a list of possible recipients $r_i \in \mathcal{R}$ together with some auxiliary information aux. The meaning of aux depends on the nature of the AC protocol. Based on the AC protocol, auxiliary information can specify the content of a message that is sent to a recipient or may contain a symbolic description of user behavior. We can think of the rows in the input table as a list of successive input to the protocol.

Definition 1 (Input tables)**.** *An* input table *D of size t over a user space $\mathcal{U}$, a recipient space $\mathcal{R}$ and an auxiliary information space Aux is an ordered table $D = (d_1, d_2, \ldots, d_t)$ of tuples $d_j = (u_j, (r_{j_i}, \mathsf{aux}_{j_i})_{i=1}^{\ell})$, where $u_j \in \mathcal{U}, r_{j_i} \in \mathcal{R}$ and $\mathsf{aux}_{j_i} \in \mathsf{Aux}$.*

A typical adversary in an AC protocol can compromise a certain number of parties. We model such an adversary capability as static corruption: before the protocol execution starts $\mathcal{A}$ may decide which parties to compromise.

Our protocol model is generic enough to capture multiparty protocols in classical simulation-based composability frameworks, such as the UC [23], the IITM [24] or the RSIM [25] framework. In particular, our protocol model comprises ideal functionalities, trusted machines that are used in simulation-based composability frameworks to define security. It is straightforward to construct a wrapper for such an ideal functionality of an AC protocol that translates input tables to the expected input of the functionality. We present such a wrapper for Tor in Section VI.

B. Generalized Computational Differential Privacy

For privacy preserving computations the notion of *differential privacy* (DP) [20], [21] is a standard for quantifying privacy. Informally, differential privacy of a mechanism guarantees that the mechanism does not leak any information about a single user–even to an adversary that has auxiliary information about the rest of the user base. It has also been generalized to protocols against computationally bounded adversaries, which has led to the notion of computational differential privacy (CDP) [22]. In computational differential privacy two input tables are compared that are *adjacent* in the sense that they only differ in one row, called the *challenge row*. The definition basically states that no PPT adversary should be able to determine which of the two input tables was used.

For anonymity properties of AC protocols, such a notion of adjacency is too strong. One of the main objectives of an AC protocol is communication: delivering the sender's message to the recipient. However, if these messages carry information about the sender, a curious recipient can determine the sender (see the following example).

Example (Privacy): Consider an adversary $\mathcal{A}$ against the "computational differential privacy" game with an AC protocol. Assume the adversary owns a recipient evilserver.com*, that forwards all messages it receives to $\mathcal{A}$. Initially, $\mathcal{A}$ sends input tables D_0, D_1 to the IND-CDP challenger that are equal in all rows but one: In this distinguishing row of D_0 the party Alice sends the message "I am Alice!" to evilserver.com and in D_1, the party Bob sends the message "I am Bob!" to evilserver.com. The tables are adjacent in the sense of computational differential privacy (they differ in exactly one row). However, no matter how well the identities of recipients are hidden by the protocol, the adversary can recognize them by their messages and thus will win the game with probability* 1.

Our generalization of CDP allows more fine-grained notions of adjacency; e.g., adjacency for sender anonymity means that the two tables only differ in one row, and in this row only the user that sends the messages is different. In general, we say that an adjacency function α is a randomized function that expects two input tables (D_0, D_1) and either outputs two input tables (D'_0, D'_1) or a distinguished error symbol $\perp$. Allowing the adjacency function α to also modify the input tables is useful for shuffling rows, which we need for defining relationship anonymity (see Definition 6).

CDP, like the original notion of differential privacy, only considers trusted mechanisms. In contrast to those incorruptible, monolithic mechanisms we consider arbitrary protocols, and thus even further generalize and strengthen CDP: we grant the adversary the possibility of compromising parties in the mechanism in order to accurately model the adversary.

For analyzing a protocol $\mathcal{P}$, we define a challenger $\textsc{Ch}(\mathcal{P}, \alpha, b)$ that expects two input tables D_0, D_1 from a PPT adversary $\mathcal{A}$. The challenger $\textsc{Ch}$ calls the adjacency function α on (D_0, D_1). If α returns $\perp$ the challenger halts. Otherwise, upon receiving two (possibly modified) tables D'_0, D'_1, $\textsc{Ch}$ chooses D'_b, depending on its input bit b, and successively feeds one row after the other to the protocol $\mathcal{P}$.[2] We assume that the protocol upon an input $(u, (r_i, \mathsf{aux}_i)_{i=1}^{\ell})$, sends $(r_i, \mathsf{aux}_i)_{i=1}^{\ell}$ as input to

[1] We stress that using standard methods, a distributed protocol with several parties can be represented by one interactive Turing machine.

[2] In contrast to IND-CDP, we only consider ppt-computable tables.

```
Upon message(input, $D_0, D_1$) (only once)
  compute $(D'_0, D'_1) \leftarrow \alpha(D_0, D_1)$
  if $(D'_0, D'_1) \neq \perp$ then
    run $\mathcal{P}$ on the input table $D'_b$ and forward all
    messages that are sent by $\mathcal{P}$ to the adversary $\mathcal{A}$
    and send all messages by the adversary to $\mathcal{P}$.
```

Figure 1. The challenger $\text{Ch}(\mathcal{P}, \alpha, b)$ for the adjacency function α

party u. In detail, upon a message $(\mathsf{input}, D_0, D_1)$ sent by $\mathcal{A}$, $\text{Ch}(\mathcal{P}, \alpha, b)$ computes $(D'_0, D'_1) \leftarrow \alpha(D_0, D_1)$. If $(D'_0, D'_1) \neq \perp$, Ch runs $\mathcal{P}$ with the input table D'_b and forwards all messages that are sent from $\mathcal{P}$ to $\mathcal{A}$ and all messages that are sent from $\mathcal{A}$ to $\mathcal{P}$. At any point, the adversary may output his decision b^*.

Now we formally define (ε, δ)-α-IND-CDP. Figure 1 shows the full construction of Ch.

Definition 2 ((ε, δ)-α-IND-CDP)**.** *Let* Ch *be the challenger from Figure 1. The protocol $\mathcal{P}$ is* (ε, δ)-α-IND-CDP for α, *where* $\varepsilon \geq 0$ *and* $0 \leq \delta \leq 1$, *if for all* PPT*-adversaries $\mathcal{A}$:*

$$\Pr[b = 0 : b \leftarrow \mathcal{A}^{\text{Ch}(\mathcal{P},\alpha,0)}] \\ \leq e^{\varepsilon} \cdot \Pr[b = 0 : b \leftarrow \mathcal{A}^{\text{Ch}(\mathcal{P},\alpha,1)}] + \delta$$

In the commonly used communication-efficient AC protocols such as Tor, $\varepsilon = 0$. However, we keep the parameter ε to maintain generality, since there are AC protocols in the literature with $\varepsilon > 0$ (e.g., pool mixes with dummy traffic [31]).

A note on the adversary model. While our adversary initially constructs the two input tables in their entirety, our model does not allow the adversary to adaptively react to the information that it observes by changing the behaviors of users. This is in line with previous work, which also assumes that the user behavior is fixed before the protocol is executed [10], [14].

As a next step towards defining our anonymity properties, we formally introduce the notion of challenge rows. Recall that challenge rows are the rows that differ in the two input tables.

Definition 3 (Challenge rows)**.** *Given two input tables $A = (a_1, a_2, \ldots, a_t)$ and $B = (b_1, b_2, \ldots, b_t)$ of the same size, we refer to all rows $a_i \neq b_i$ with $i \in \{1, \ldots, t\}$ as* challenge rows*. If the input tables are of different sizes, there are no challenge rows. We denote the challenge rows of D as $\mathsf{CR}(D)$.*

C. Anonymity properties

In this section, we present our (ε, δ)-α-IND-CDP based anonymity definitions in which the adversary is allowed to choose the entire communication except for the challenge rows, for which he can specify two possibilities. First, we define sender anonymity, which states that a malicious recipient cannot decide, for two candidates, to whom he is talking even in the presence of virtually arbitrary auxiliary information. Second, we define user unlinkability, which states that a malicious recipient cannot decide whether it is communicating with one user or with two different users, in particular even if he chooses the two possible rows. Third, we define relationship anonymity, which states that an adversary (that potentially controls some protocol parties) cannot relate sender and recipient in a communication.

```
$\alpha_{\text{SA}}(D_0, D_1)$
  if $||D_0|| \neq ||D_1||$ then
    output $\perp$
  if $\mathsf{CR}(D_0) = ((u_0, R)) \wedge \mathsf{CR}(D_1) = ((u_1, R))$ then
    output $(D_0, D_1)$
  else
    output $\perp$
```

Figure 2. The adjacency function α_{SA} for sender anonymity.

Our definitions are parametrized by ε and δ. We stress that all our definitions are necessarily quantitative. Due to the adversary's capability to compromise parts of the communication network and the protocol parties, achieving overwhelming anonymity guarantees (i.e., for a negligible δ) for non-trivial (and useful) AC protocols is infeasible.

1) Sender anonymity: Sender anonymity requires that the identity of the sender is hidden among the set of all possible users. In contrast to other notions from the literature, we require that the adversary is not able to decide which of two *self-chosen* users have been communicating. Our notion is stronger than the usual notion, and in Section IV we exactly quantify the gap between our notion and the notion from the literature. Moreover, we show that the Tor network satisfies this strong notion, as long as the user in question did not choose a compromised path (see Section VI).

We formalize our notion of sender anonymity with the definition of an adjacency function α_{SA} as depicted in Figure 2. Basically, α_{SA} merely checks whether in the challenge rows everything except for the user is the same.

Definition 4 (Sender anonymity)**.** *A protocol $\mathcal{P}$ provides* (ε, δ)-sender anonymity *if it is* (ε, δ)-α-IND-CDP *for* α_{SA} *as defined in Figure 2.*

Example (Sender anonymity): The adversary $\mathcal{A}$ decides that he wants to use users Alice and Bob in the

```
$\alpha_{\text{UL}}(D_0, D_1)$
    if $||D_0|| \neq ||D_1||$ then
        output $\perp$
    if $\mathsf{CR}(D_0) = ((u_0, R_u), (u_0, R_v)) =: (c_{0,u}, c_{0,v})$
     $\wedge \mathsf{CR}(D_1) = ((u_1, R_u), (u_1, R_v)) =: (c_{1,u}, c_{1,v})$
    then
        $x \xleftarrow{R} \{0, 1\}$, $y \xleftarrow{R} \{u, v\}$
        Replace $c_{x,y}$ with $c_{(1-x),y}$ in $D_x$
        output $(D_x, D_{1-x})$
    else
        output $\perp$
```

Figure 3. The adjacency function α_{UL} for sender unlinkability.

```
$\alpha_{\text{Rel}}(D_0, D_1)$
    if $||D_0|| \neq ||D_1||$ then
        output $\perp$
    if $\mathsf{CR}(D_0) = ((u_0, R_u)) \wedge \mathsf{CR}(D_1) = ((u_1, R_v))$
    then
        $x \xleftarrow{R} \{0, 1\}$, $y \xleftarrow{R} \{0, 1\}$
        if x=1 then
            Set $\mathsf{CR}(D_0)$ to $((u_1, R_v))$
        if y=1 then
            Set $\mathsf{CR}(D_1)$ to $((u_0, R_v))$
        else
            Set $\mathsf{CR}(D_1)$ to $((u_1, R_u))$
        output $(D_0, D_1)$
    else
        output $\perp$
```

Figure 4. The adjacency function α_{Rel} for relationship anonymity.

sender anonymity game. It sends input tables D_0, D_1 such that in the challenge row of D_0 Alice sends a message m^* of $\mathcal{A}$'s choice to a (probably corrupted) recipient, e.g. evilserver.com, and in D_1, instead of Alice, Bob sends the same message m^* to the same recipient evilserver.com. The adjacency function α_{Rel} makes sure that only one challenge row exists and that the messages and the recipients are equal. If so, it outputs D_0, D_1 and if not it outputs $\perp$.

Notice that analogously recipient anonymity (α_{RA}) can be defined: the adjacency function then checks that the challenge rows only differ in one *recipient*.

2) Sender unlinkability: A protocol satisfies *sender unlinkability*, if for any two actions, the adversary cannot determine whether these actions are executed by the same user [32]. We require that the adversary does not know whether two challenge messages come from the same user or from different users. We formalize this intuition by letting the adversary send two input tables with two challenge rows, respectively. Each input table D_x carries challenge rows in which a user u_x sends a message to two recipients R_u, R_v. We use the shuffling abilities of the adjacency function α_{UL} as defined in Figure 3, which makes sure that D'_0 will contain the same user in both challenge rows, whereas D'_1 will contain both users. As before, we say a protocol $\mathcal{P}$ fulfills sender unlinkability, if no adversary $\mathcal{A}$ can sufficiently distinguish $\textsc{Ch}(\mathcal{P}, \alpha_{\text{UL}}, 0)$ and $\textsc{Ch}(\mathcal{P}, \alpha_{\text{UL}}, 1)$. This leads to the following concise definition.

Definition 5 (Sender unlinkability)**.** *A protocol $\mathcal{P}$ provides* (ε, δ)-sender unlinkability *if it is* (ε, δ)-α-IND-CDP *for α_{UL} as defined in Figure 3.*

Example (Sender unlinkability): The adversary $\mathcal{A}$ decides that he wants to use users Alice and Bob in the unlinkability game. He sends input tables D_0, D_1 such that in the challenge rows of D_0 Alice sends two messages to two recipients and in D_1, Bob sends the same two messages to the same recipients. Although initially "the same user sends the messages" would be true for both input tables, the adjacency function α_{UL} changes the challenge rows in the two input tables D_0, D_1. In the transformed input tables D'_0, D'_1, only one of the users (either Alice or Bob) will send both messages in D'_0 , whereas one message will be sent by Alice and the other by Bob in D'_1.

3) Relationship anonymity: $\mathcal{P}$ satisfies *relationship anonymity*, if for any action, the adversary cannot determine sender and recipient of this action at the same time [32]. We model this property by letting the adjacency α_{Rel} check whether it received an input of two input tables with a single challenge row. We let the adjacency function α_{Rel} shuffle the recipients and sender such that we obtain the four possible combinations of user and recipient. If the initial challenge rows are (u_0, R_0) and (u_1, R_1), α_{Rel} will make sure that in D'_0 one of those initial rows is used, where in D'_1 one of the rows (u_0, R_1) or (u_1, R_0) is used.

We say that $\mathcal{P}$ fulfills relationship anonymity, if no adversary can sufficiently distinguish $\textsc{Ch}(\mathcal{P}, \alpha_{\text{Rel}}, 0)$ and $\textsc{Ch}(\mathcal{P}, \alpha_{\text{Rel}}, 1)$.

Definition 6 (relationship anonymity)**.** *A protocol $\mathcal{P}$ provides* (ε, δ)-relationship anonymity *if it is* (ε, δ)-α-IND-CDP *for α_{Rel} as defined in Figure 4.*

Example (Relationship anonymity): The adversary $\mathcal{A}$ decides that he wants to use users Alice and Bob and the recipients Charly and Eve in the relationship anonymity game. He wins the game if he can distinguish between the scenario "0" where Alice sends m_1 to Charly or Bob sends m_2 to Eve and the scenario "1" where Alice sends

```
Upon message $(\mathsf{input}, D)$ (only once)
  if $\exists!$ challenge row in $D$ then
    Place user $u$ in the challenge row of $D$
    run $\mathcal{P}$ on the input table $D$ and forward all messages to $\mathcal{A}$
```

Figure 5. The challenger $\mathrm{SACh}(\mathcal{P}, u)$

m_2 to Eve or Bob sends m_1 to Charly. Only one of those four possible input lines will be fed to the protocol.

$\mathcal{A}$ sends input tables D_0, D_1 such that in the challenge row of D_0 Alice sends m_1 to Charly and in D_1, Bob sends m_2 to Eve. Although initially 'scenario 0" would be true for both input tables, the adjacency function α_{Rel} changes the challenge rows in the two input tables D_0, D_1 such that in D_0' one of the two possible inputs for scenario "0" will be present (either Alice talks to Charly or Bob talks to Eve) and in D_1' one of the two possible inputs for scenario "1" will be present (either Bob talks to Charly or Alice talks to Eve).

IV. Studying our anonymity definitions

In this section, we show that our anonymity definitions indeed capture the anonymity notions from the literature. We compare our notions to definitions that are directly derived from informal descriptions in the seminal work by Pfitzmann and Hansen [32]. Lastly, we investigate the relation between our own anonymity definitions.

A. Sender anonymity

The notion of sender anonymity is introduced in [32] as follows:

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

From this description, we formalize their notion of sender anonymity. For any message m and adversary $\mathcal{A}$, any user in the user space is equally likely to be the sender of m.

Definition 7 (δ-sender anonymity)**.** *A protocol $\mathcal{P}$ with user space $\mathcal{U}$ of size N has* δ-sender anonymity *if for all* PPT*-adversaries $\mathcal{A}$*

$$Pr\left[u^* = u : u^* \leftarrow \mathcal{A}^{\mathrm{SACh}(\mathcal{P},u)}, u \xleftarrow{R} \mathcal{U}\right] \leq \frac{1}{N} + \delta,$$

where the challenger SACh *as defined as in Figure 5.*

Note that SACh slightly differs from the challenger $\mathrm{Ch}(\mathcal{P}, \alpha, b)$ in Figure 1: It does not require two, but just one input table in which a single row misses its sender. We call this row the challenge row.

This definition is quite different from our interpretation with adjacency functions. While α_{SA} requires $\mathcal{A}$ to simply distinguish between two possible outcomes, Definition 7 requires $\mathcal{A}$ to correctly guess the right user. Naturally, α_{SA} is stronger than the definition above. Indeed, we can quantify the gap between the definitions: Lemma 8 states that an AC protocol satisfies $(0, \delta)$-α_{SA} implies that this AC also has δ-sender anonymity. The proofs for these lemmas can be found in the extended version [33]. In this section, we only present the proof outlines.

Lemma 8 (sender anonymity)**.** *For all protocols $\mathcal{P}$ over a (finite) user space $\mathcal{U}$ of size N it holds that if $\mathcal{P}$ has $(0, \delta)$-α-IND-CDP for α_{SA}, $\mathcal{P}$ also has δ-sender anonymity as in Definition 7.*

Proof outline: We show the contraposition of the lemma: an adversary A that breaks sender anonymity, can be used to break α-IND-CDP for α_{SA}. We construct an attacker B against α-IND-CDP for α_{SA} by choosing the senders of the challenge rows at random, running A on the resulting game, and outputting the same as A. For A the resulting view is the same as in the sender anonymity game; hence, B has the same success probability in the α-IND-CDP game as A in the sender anonymity game. ■

In the converse direction, we lose a factor of $\frac{1}{N}$ in the reduction, where N is the size of the user space. If an AC protocol $\mathcal{P}$ provides δ-sender anonymity, we only get $(0, \delta \cdot N)$-α_{SA} for $\mathcal{P}$.

Lemma 9. *For all protocols $\mathcal{P}$ over a (finite) user space $\mathcal{U}$ of size N it holds that if $\mathcal{P}$ has δ-sender anonymity as in Definition 7, $\mathcal{P}$ also has $(0, \delta \cdot N)$-α-IND-CDP for α_{SA}.*

Proof outline: We show the contraposition of the lemma: an adversary A that breaks α-IND-CDP for α_{SA}, can be used to break sender anonymity. We construct an attacker B against sender anonymity by running A on the sender anonymity game and outputting the same as A. If the wishes of A for the challenge senders coincide with the sender that the challenger chose at random, the resulting view is the same as in the α-IND-CDP game for α_{SA}; hence, B has a success probability of δ/N in the sender anonymity game if A has a success probability of δ in the α-IND-CDP game for α_{SA}. ■

B. Unlinkability

The notion of unlinkability is defined in [32] as follows:

> Unlinkability of two or more items of interest (IOIs, e.g., subjects, messages, actions, ...) from an attacker's perspective means that within the system (comprising these and possibly other

```
**Upon message** (input, $D$) **(only once)**
  **if** exactly 2 rows in $D$ are missing the user **then**
    $u_0 \xleftarrow{R} \mathcal{U}, u_1 \xleftarrow{R} \mathcal{U} \setminus \{u_0\}$
    **if** $b = 0$ **then**
      Place $u_0$ in both rows.
    **else**
      Place $u_0$ in the first and $u_1$ in the second row.
    run $\mathcal{P}$ on input table $D$ and forward all messages
    to $\mathcal{A}$
```

Figure 6. The challenger ULCH($\mathcal{P}, b$)

items), the attacker cannot sufficiently distinguish whether these IOIs are related or not.

Again, we formalize this in our model. We leave the choice of potential other items in the system completely under adversary control. Also, the adversary controls the "items of interest" (IOI) by choosing when and for which recipient/messages he wants to try to link the IOIs. Formally, we define a game between a challenger ULCH and an adversary $\mathcal{A}$ as follows: First, $\mathcal{A}$ chooses a input table D, but leaves the place for the users in two rows blank. The challenger then either places one (random) user in both rows or two different (random) users in each and then runs the protocol and forwards all output to $\mathcal{A}$. The adversary wins the game if he is able to distinguish whether the same user was placed in the rows (i.e. the IOIs are linked) or not.

Definition 10 (δ-sender unlinkability). *A protocol $\mathcal{P}$ with user space $\mathcal{U}$ has δ-*sender unlinkability *if for all* PPT-*adversaries $\mathcal{A}$*

$$\Big| Pr\left[b = 0 : b \leftarrow \mathcal{A}^{\mathrm{ULCH}(\mathcal{P},0)}\right] - Pr\left[b = 0 : b \leftarrow \mathcal{A}^{\mathrm{ULCH}(\mathcal{P},1)}\right] \Big| \leq \delta$$

where the challenger ULCH *is as defined in Figure 6.*

We show that our notion of sender unlinkability using the adjacency function α_{UL} is much stronger than the δ-sender unlinkability Definition 10: $(0, \delta)$-α_{UL} for an AC protocol directly implies δ-sender unlinkability; we do not lose any anonymity.

Lemma 11 (sender unlinkability). *For all protocols $\mathcal{P}$ over a user space $\mathcal{U}$ it holds that if $\mathcal{P}$ has $(0, \delta)$-α-*IND-CDP *for α_{UL}, $\mathcal{P}$ also has δ-sender unlinkability as in Definition 10.*

Proof outline: We show the contraposition of the lemma: an adversary A that breaks sender unlinkability, can be used to break α-IND-CDP for α_{UL}. We construct an attacker B against α-IND-CDP for α_{UL} by choosing the senders of the challenge rows at random, running A on the resulting game, and outputting the same as A. For A the resulting view is the same as in the sender unlinkability game; hence, B has the same success probability in the α-IND-CDP game for α_{UL} as A in the sender unlinkability game. ■

For the converse direction, however, we lose a factor of roughly N^2 for our δ. Similar to above, proving that a protocol provides δ-sender unlinkability only implies that the protocol is $(0, \delta \cdot N(N-1))$-α-IND-CDP for α_{UL}.

Lemma 12 (sender unlinkability). *For all protocols $\mathcal{P}$ over a user space $\mathcal{U}$ of size N it holds that if $\mathcal{P}$ has δ-sender unlinkability as in Definition 10, $\mathcal{P}$ also has* $(0, \delta \cdot N(N-1))$-α-IND-CDP *for α_{UL}.*

Proof outline: We show the contraposition of the lemma: an adversary A that breaks α-IND-CDP for α_{UL}, can be used to break sender unlinkability. We construct an attacker B against sender unlinkability by running A on the sender unlinkability game and outputting the same as A. If the senders from the challenge from of A coincide with the senders that the challenger chose at random, the resulting view is the same as in the α-IND-CDP game for α_{UL}; hence, B has a success probability of $\delta / N(N-1)$ in the sender unlinkability game if A has a success probability of δ in the α-IND-CDP game for α_{UL}. ■

C. Relationship anonymity

While for sender anonymity and sender unlinkability our notions coincide with the definitions used in the literature, we find that for relationship anonymity, many of the interpretations from the literature are not accurate. In their Mixnet analysis, Shmatikov and Wang [8] define relationship anonymity as 'hiding the fact that party A is communicating with party B'. Feigenbaum et al. [11] 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 [32]: 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. if he breaks sender anonymity), also breaks the relationship anonymity game, all sender-recipient pairs are no longer equally likely. Therefore we think that approach via the adjacency function gives a better definition of relationship

$$\begin{tikzcd}
\alpha_{\mathrm{UL}} \arrow[r, Rightarrow, bend right, "/" marking] & \alpha_{\mathrm{SA}} \arrow[l, Rightarrow, bend right] \arrow[r, Rightarrow, bend left] & \alpha_{\mathrm{Rel}} \arrow[l, Rightarrow, bend left, "/" marking]
\end{tikzcd}$$

Figure 7. The relations between our anonymity definitions

anonymity because the adversary needs to uncover both sender and recipient in order to break anonymity.

D. Relations between anonymity notions

Having justified the accuracy of our anonymity notions, we proceed by presenting the relations between our notions of anonymity. AnoA allows us to formally argue about these relations. Figure 7 illustrates the implications we get based on our definitions using adjacency functions. In this section, we discuss these relations. The proofs can be found in the extended version [33].

Lemma 13 (Sender anonymity implies relationship anonymity.). *If a protocol $\mathcal{P}$ has* $(0, \delta)$-α-IND-CDP *for* α_{SA}*, is also has* $(0, \delta)$-α-IND-CDP *for* α_{Rel}*.*

Proof outline: Relationship anonymity requires an adversary to acquire information about both sender and recipient. If a protocol has sender anonymity, this is not possible. Hence, sender anonymity implies relationship anonymity. ■

Similarly, recipient anonymity implies relationship anonymity.

Lemma 14 (Sender anonymity implies sender unlinkability). *If a protocol $\mathcal{P}$ has* $(0, \delta)$-α-IND-CDP *for* α_{SA}*, $\mathcal{P}$ also has* $(0, \delta)$-α-IND-CDP *for* α_{UL}*.*

Proof outline: Our strong adversary can determine the behavior of all users; in other words, the adversary can choose the scenario in which it wants to deanonymize the parties in question. Thus, the adversary can choose the payload messages that are not in the challenge row such that these payload messages leak the identity of their sender. Hence, if an adversary can link the message in the challenge row to another message, it can determine the sender. Thus, sender anonymity implies sender unlinkability. ■

A protocol could leak the sender of a single message. Such a message does not necessarily help an adversary in figuring out whether another message has been sent by the same sender, but breaks sender anonymity.

Lemma 15 (Sender unlinkability does not imply sender anonymity). *If a protocol $\mathcal{P}$ has* $(0, \delta)$-α-IND-CDP *for* α_{UL}*, $\mathcal{P}$ does not necessarily have* $(0, \delta')$-α-IND-CDP *for* α_{SA} *for any $\delta' < 1$.*

Proof outline: We consider a protocol Π that satisfies sender anonymity. We, moreover, consider the modified protocol Π' that leaks the sender of a single message. Since by Lemma 14 Π satisfies unlinkability, we conclude that the modified protocol Π' satisfies sender unlinkability: a single message does not help the adversary in breaking sender unlinkability. However, Π' leaks in one message the identity of the sender in plain, hence does not satisfy sender anonymity. ■

Relationship anonymity does not imply sender anonymity in general: for example, a protocol may reveal information about senders of the messages, but not about recipients or message contents.

Lemma 16 (Relationship anonymity does not imply sender anonymity). *If a protocol $\mathcal{P}$ has* $(0, \delta)$-α-IND-CDP *for* α_{Rel}*, $\mathcal{P}$ does not necessarily have* $(0, \delta')$-α-IND-CDP *for* α_{SA} *for any $\delta' < 1$.*

Proof outline: We consider a protocol Π that satisfies sender anonymity. We, moreover, consider the modified protocol Π' that for each message leaks the sender. Since by Lemma 13 Π satisfies unlinkability, we conclude that the modified protocol Π' satisfies relationship anonymity: the sender alone does not help the adversary in breaking relationship anonymity. However, Π' leaks the identity of the sender in plain, hence does not satisfy sender anonymity. ■

This concludes the formal definition of our framework.

V. Leveraging UC realizability

Our adversary model in AnoA is strong enough to capture well-known simulation-based composability frameworks (e.g., UC [23], IITM [24] or RSIM [25]). In Section VI we apply AnoA to a model in the simulation-based universal composability (UC) framework.

In this section, we briefly introduce the UC framework and then prove that α-IND-CDP is preserved under realization. Moreover, we discuss how this preservation allows for an elegant crypto-free anonymity proof for cryptographic AC protocols.

A. The UC framework

The UC framework allows for a modular analysis of security protocols. In the framework, the security of a protocol is defined by comparing it 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*.

Security in the UC framework is defined as follows: A protocol is secure if an execution of this protocol is indistinguishable from an execution of the corresponding ideal functionality.

More formally, the notion of indistinguishability is captured in UC in terms of *realization*: A protocol π *UC-realizes* an ideal functionality $\mathcal{F}$ if for all PPT adversaries $\mathcal{A}$ there is a PPT simulator S such that no PPT machine can distinguish an interaction with π and $\mathcal{A}$ from an interaction with $\mathcal{F}$ and S. The distinguisher is connected to the protocol and the adversary (or the simulator). A full definition can be found in the extended version [33].

B. *Preservation of* α-IND-CDP

We prove that α-IND-CDP is preserved by UC realization. This result is motivated by the ideas presented in the result of integrity property conservation by simulation-based indistinguishability shown by Backes and Jacobi [34, Thm. 1].

As a consequence of this lemma, it suffices to apply AnoA to ideal functionalities: transferring the results to the real protocol weakens the anonymity guarantees only by a negligible amount.

Lemma 17 (Preservation lemma)**.** *Let $\mathcal{P}$ be (ϵ, δ)-α-IND-CDP and Π be a protocol. If Π UC-realizes $\mathcal{P}$ then Π is (ϵ, Δ)-α-IND-CDP with $\Delta = \delta + \delta'$ for some negligible value δ'.*

Proof outline: The proof of the preservation lemma is straightforward: If the success probability of an adversary in the real world differs in more than a negligible value from the ideal world, we can use this adversary to distinguish the real from the ideal game. ■

The full proof can be found in the extended version [33]. This preservation lemma, in combination with an ideal functionality for an AC protocol, is useful for analyzing the AC protocol with respect to our strong anonymity definitions. In the next section, we exemplify approach by using an ideal functionality for Tor [30] and showing that the anonymity analysis of Tor boils down to a purely combinatorical analysis.

VI. Analyzing Tor Anonymity

The onion routing (OR) [18] network Tor [1] is the most successful anonymity technology to date: hundreds of thousands individuals all over the world use it today to protect their privacy over the Internet. Naturally, Tor is our first choice for applying our AnoA framework.

We start our discussion by briefly describing the Tor protocol [35] and its UC definition [30]. We then formally prove (ε, δ)-α-IND-CDP for Tor's UC definition and quantify anonymity provided by the Tor network in terms of the anonymity properties defined in Section III. Finally, we consider a selection of system-level attacks (e.g., traffic analysis) and adaptations (e.g., entry guards) for Tor, and analyze their effects on Tor's anonymity guarantees.

A. *Tor—The OR Network*

An OR network such as Tor [35] consists of a set of *OR nodes (or proxies)* that relay traffic, a large set of users and a directory service that maintains and provides cryptographic and routing information about the OR nodes. Users utilize the Tor network by selecting a sequence of OR nodes and creating a path, called a *circuit*, over this set. This circuit is then used to forward the users' traffic and obscure the users' relationship with their destinations. It is important that an OR node cannot determine the circuit nodes other than its immediate predecessor and successor. In the OR protocol, this is achieved by wrapping every message in multiple layers of symmetric-key encryption. Symmetric keys are agreed upon between each OR node in the circuit and the user during the circuit construction phase.

Tor was designed to guarantee anonymity against partially global attackers, i.e., attackers that do not only control some OR nodes but also a portion of the network. However, an accurate anonymity quantification is not possible without formally modeling the OR protocol and its adversary. In an earlier work, Backes et al. [30] presented a formal UC definition (an ideal functionality $\mathcal{F}_{\text{OR}}$) for the OR network, and proposed a practical cryptographic instantiation which is currently employed in the Tor network. We employ this ideal functionality $\mathcal{F}_{\text{OR}}$ for instantiating the AnoA framework.

B. *Anonymity Analysis*

We start our Tor analysis with a brief overview of the $\mathcal{F}_{\text{OR}}$ functionality and refer the readers to [30] for more details. An excerpt of relevant details can also be found in the extended version [33]. $\mathcal{F}_{\text{OR}}$ 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. $\mathcal{F}_{\text{OR}}$ 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 $\mathcal{F}_{\text{OR}}$ these secure channels are realized by having each party store their messages in the shared memory, and create and send corresponding handles $\langle P, P_{next}, h\rangle$ through the network. Here, P and P_{next} are the sender and the recipient of a message respectively and h is a handle, or pointer, for the message

```
Upon input $(r_i, m_i)_{i=0}^{\ell}$
  $\mathcal{P} \leftarrow \mathrm{RandomParties}(P_u)$
  send message $(\mathsf{cc}, \mathcal{P})$ to $P_u$
  wait for response $(\mathsf{created}, \mathcal{C})$
  for all $(r_i, m_i), i \in \{1, \dots, \ell\}$ do
    send message $(\mathsf{send}, \mathcal{C}, m_i)$ to $P_u$

$\mathrm{RandomParties}(P_u)$:
  $l \xleftarrow{R} \{1, \dots, n\}$
  $N := \{1, \dots, n\}$
  for $j = 1$ to $l$ do
    $i_j \xleftarrow{R} N$
    $N := N \setminus \{i_j\}$
  return $(P_u, P_{i_1}, \dots, P_{i_l})$
```

Figure 8. Wrapper module $\textsc{Env}_u$ for onion proxy P_u

```
Upon message $m$ from $\mathcal{F}_{\text{NET}}$ or $\mathcal{F}_{\text{OR}}$
  send $m$ to the challenger
  reflect the message $m$ back to sender
```

Figure 9. Dummy-adversary in $\mathcal{F}_{\text{OR}}$

in the shared memory. Only messages that are visible to compromised parties are forwarded to $\mathcal{A}$.

We consider a partially global, passive adversary for our analysis using AnoA, i.e., $\mathcal{A}$ decides on a subset of nodes before the execution, which are then compromised. The adversary $\mathcal{A}$ then only reads intercepted messages, but does not react to them.

Tor sets a time limit (of ten minutes) for each established circuit. However, the UC framework does not provide a notion of time. $\mathcal{F}_{\text{OR}}$ models such a time limit by only allowing a circuit C to transport at most a constant number (say ttl_C) of messages.

In the context of onion routing, we interpret an input table $D = (d_1, d_2, \dots, d_t)$ as follows: each row $d_i = (u, (r_j, \mathsf{aux}_j)_{j=1}^{\ell})$ defines a session transmitted through the OR-network, where aux_j is the message sent from the user to recipient r_j. An input table thus defines a sequence of sessions sent through the OR-network.

We assume that for each row in an input table, a new circuit in the OR-network is drawn, as each row defines a newly started OR session. Furthermore, the number of messages per row (or session) is bounded by ttl_C.

In order to make $\mathcal{F}_{\text{OR}}$ compatible with our α-IND-CDP definition, we require an additional wrapper functionality, which processes the input rows forwarded from the challenger Ch. This functionality is defined in Figure 8. $\textsc{Env}_u$ receives a row, which had u as its user, as its input. It then initiates the circuit construction for the new session and sends all messages in the row through this circuit.

Messages intercepted by compromised nodes are sent to a network adversary $\mathcal{F}_{\text{NET}}$ described in Fig. 9. $\mathcal{F}_{\text{NET}}$ forwards all intercepted messages to the challenger, who in turn forwards them to $\mathcal{A}$.

We show that the Tor analysis can be based on a *distinguishing event* $\mathcal{D}$, which has already been identified in the first onion routing anonymity analysis by Syverson et al. [2, Fig. 1]. The key observation is that the adversary can only learn about the sender or recipient of some message if he manages to compromise the entry- or exit-node of the circuit used to transmit this message. We define the distinguishing event $\mathcal{D}_\alpha$ for each of the anonymity notions defined in section III.

Sender Anonymity (α_{SA}). Let $\mathcal{D}_{\alpha_{\text{SA}}}$ be the event that the entry-node of the challenge row is compromised by $\mathcal{A}$. This allows $\mathcal{A}$ to determine the sender of the challenge row and therefore break sender anonymity.

Sender Unlinkability (α_{UL}). Let $\mathcal{D}_{\alpha_{\text{UL}}}$ be the event that $\mathcal{A}$ successfully compromises the entry nodes for both challenge rows in the unlinkability game. This allows $\mathcal{A}$ to determine whether the sessions defined by the challenge rows are linked or not, and hence break the unlinkability game.

Relationship Anonymity (α_{Rel}). Let $\mathcal{D}_{\alpha_{\text{Rel}}}$ be the event that $\mathcal{A}$ successfully compromises entry- and exit-node of the challenge row. This allows him to link both sender and recipient of the sessions associated with the challenge row.

We first prove Lemma 18. It captures anonymity provided by $\mathcal{F}_{\text{OR}}$ in case $\mathcal{D}$ does not happen. We then use Lemma 18 to prove (ε, δ)-α-IND-CDP for $\mathcal{F}_{\text{OR}}$ in general.

We introduce random strings $r_{\mathcal{A}}$ and r_{Ch} as additional input to the adversary and the challenger respectively. This allows us to handle them as deterministic machines and simplifies the proof for Lemma 18. Accordingly, all subsequent probabilities are taken over those random strings. In the following, we present a proof outline here. The full proof is available in the extended version [33].

Lemma 18. *Let $r_{\mathcal{A}}, r_{\text{Ch}} \xleftarrow{R} \{0,1\}^{p(\eta)}$. Given two input tables D_1, D_0 which are adjacent for $\alpha \in \{\alpha_{\text{SA}}, \alpha_{\text{UL}}, \alpha_{\text{Rel}}\}$, it holds that*

$$\begin{aligned}&\Pr[\mathcal{A}^{\text{Ch}(\mathcal{F}_{\text{OR}}, \alpha, 0, r_{\text{Ch}})}(r_{\mathcal{A}}) = 0 \mid \neg\mathcal{D}_\alpha(r_{\text{Ch}}, r_{\mathcal{A}})]\\ =&\Pr[\mathcal{A}^{\text{Ch}(\mathcal{F}_{\text{OR}}, \alpha, 1, r_{\text{Ch}})}(r_{\mathcal{A}}) = 0 \mid \neg\mathcal{D}_\alpha(r_{\text{Ch}}, r_{\mathcal{A}})]\end{aligned}$$

Proof outline: We fix the random string r_{Ch}. This in turn fixes the circuits drawn by $\mathcal{F}_{\text{OR}}$ for each row. As circuits are drawn independently from the transmitted messages, $\mathcal{F}_{\text{OR}}$ draws the same set of circuits to transmit either input table.

We assume the event $\neg\mathcal{D}_\alpha$. For α_{SA} and α_{UL}, the messages intercepted by $\mathcal{A}$ do not carry critical information and look the same, regardless of which input table was chosen by the challenger. If we now also fix $r_\mathcal{A}$, $\mathcal{A}$ returns the same value after processing the set of intercepted messages, for either input table.

For α_{Rel}, $\mathcal{A}$ might learn partial information. But there are always at least two of the four input tables, each of which could have only been chosen by one of the challengers, for which the intercepted messages are consistent. Again, if we fix $r_\mathcal{A}$, $\mathcal{A}$ will return the same value, regardless of which challenger he is interacting with. Hence we get

$$\begin{aligned}&\Pr[\mathcal{A}^{\text{CH}(\mathcal{F}_{\text{OR}},\alpha,0,r_{\text{CH}})}(r_\mathcal{A}) = 0 \mid \neg\mathcal{D}(r_{\text{CH}}, r_\mathcal{A}), r_{\text{CH}}]\\ =&\Pr[\mathcal{A}^{\text{CH}(\mathcal{F}_{\text{OR}},\alpha,1,r_{\text{CH}})}(r_\mathcal{A}) = 0 \mid \neg\mathcal{D}(r_{\text{CH}}, r_\mathcal{A}), r_{\text{CH}}]\end{aligned}$$

and from this

$$\begin{aligned}&\Pr[\mathcal{A}^{\text{CH}(\mathcal{F}_{\text{OR}},\alpha,0,r_{\text{CH}})}(r_\mathcal{A}) = 0 \mid \neg\mathcal{D}(r_{\text{CH}}, r_\mathcal{A})]\\ =&\Pr[\mathcal{A}^{\text{CH}(\mathcal{F}_{\text{OR}},\alpha,1,r_{\text{CH}})}(r_\mathcal{A}) = 0 \mid \neg\mathcal{D}(r_{\text{CH}}, r_\mathcal{A})]\end{aligned}$$

as required. ■

With this result we obtain (ε, δ)- α-IND-CDP for $\mathcal{F}_{\text{OR}}$ by simple manipulation of equations.

Theorem 19. *$\mathcal{F}_{\text{OR}}$ is $(0, \delta)$ - α-IND-CDP for $\alpha \in \{\alpha_{\text{SA}}, \alpha_{\text{UL}}, \alpha_{\text{Rel}}\}$, i.e*

$$\begin{aligned}&\Pr[\mathcal{A}^{\text{CH}(\mathcal{F}_{\text{OR}},\alpha,0,r_{\text{CH}})}(r_\mathcal{A}) = 0]\\ \leq&\Pr[\mathcal{A}^{\text{CH}(\mathcal{F}_{\text{OR}},\alpha,1,r_{\text{CH}})}(r_\mathcal{A}) = 0]) + \delta\end{aligned}$$

with $\delta = \Pr[\mathcal{D}_\alpha(r_{\text{CH}}, r_\mathcal{A})]$.

Here δ is exactly the probability for the event $\mathcal{D}_\alpha$ that allows $\mathcal{A}$ to distinguish between both input tables. Interestingly, we get the parameter value $\varepsilon = 0$. This implies that as long as $\mathcal{D}_\alpha$ does not happen, $\mathcal{F}_{\text{OR}}$ provides perfect anonymity for its users.

C. Anonymity Quantification

We now evaluate the guarantees provided by Theorem 19 and consider further results we can derive from it for the special case of sender anonymity.

1) Distinguishing events: We measure the probability of the distinguishing event $\mathcal{D}$ using combinatorial observations. For an OR network of n OR nodes such that k of those are compromised, probabilities associated with the various anonymity notions are as follows:

Sender Anonymity (α_{SA}). The probability that $\mathcal{D}_{\alpha_{\text{SA}}}$ happens and sender anonymity is broken, is

$$Pr[\mathcal{D}_{\alpha_{\text{SA}}}] = 1 - \frac{\binom{n-1}{k}}{\binom{n}{k}} = \frac{k}{n}$$

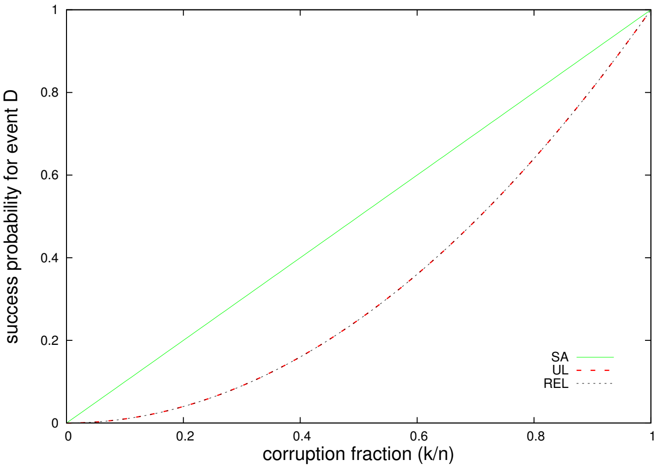

Figure 10. Probability of $\mathcal{D}$ for the different anonymity notions, depending on the corruption $\frac{k}{n}$ for 3000 OR nodes

Sender Unlinkability (α_{UL}). The probability that $\mathcal{D}_{\alpha_{\text{UL}}}$ happens and sender unlinkability is broken, is

$$Pr[\mathcal{D}_{\alpha_{\text{UL}}}] = \left(\frac{k}{n}\right)^2$$

Relationship Anonymity (α_{Rel}). The probability that $\mathcal{D}_{\alpha_{\text{Rel}}}$ happens and relationship anonymity is broken, is

$$Pr[\mathcal{D}_{\alpha_{\text{Rel}}}] = \frac{\binom{n-2}{k-2}}{\binom{n}{k}} = \frac{k(k-1)}{n(n-1)}$$

Figure 10 illustrates these results. The graph shows the probability of the distinguishing events depending on the fraction $\frac{k}{n}$ of corrupted OR nodes. We assume a system with 3000 OR nodes, which is consistent with current numbers in the real world Tor network [36]. We observe that the success probability of event $\mathcal{D}_{\alpha_{\text{SA}}}$ always remains above the success probabilities of events $\mathcal{D}_{\alpha_{\text{UL}}}$ and $\mathcal{D}_{\alpha_{\text{Rel}}}$. Therefore, sender anonymity is indeed a stronger notion than both relationship anonymity and sender unlinkability, and correspondingly more difficult to achieve. Moreover, for the usually assumed 20% corruption, the adversary's success probabilities are small for all three anonymity properties.

Note that the above analysis and the underlying model assume all OR nodes to be identical, and can perform all roles. Respecting OR node operators' legal boundaries, the real-world Tor network allows OR nodes to function in specific roles. To some extent, this simplifies $\mathcal{A}$'s task of identifying entry- or exit-nodes for circuits.

2) Multiple Challenge Rows: Considering more than one challenge row will be necessary if we want to know how the anonymity of a single user changes if he uses

Tor for more than one session or we want to consider the anonymity of a group of people which act as one entity.

If we want to extend our (ε, δ)- α-IND-CDP result w.r.t α_{SA} for Tor to more than one challenge row, we can use the direct amplification approach also known from differential privacy analysis [37]: Given two input tables D and D' with d challenge rows, we create $d-1$ intermediate input tables D_i, such that D and D_1, D_{d-1} and D' and D_i and D_{i+1} are adjacent. Repeatedly applying Theorem 19, we get the same result as for the adjacent case, but with $\delta = d \cdot Pr[\mathcal{D}_{\alpha_{\mathrm{SA}}}]$.

We can do better by realizing that the only thing that changes compared to our original analysis is the distinguishing event: Whereas before the adversary $\mathcal{A}$ could compromise only a single entry node, he now has up to d entry nodes at his disposal.

Let m be the number of distinct entry nodes used during the execution and let $\mathcal{D}^*_{\alpha_{\mathrm{SA}}}$ be the event that one of those m nodes is compromised. As noted above $m \leq d$. The probability for $\mathcal{D}^*_{\alpha_{\mathrm{SA}}}$ happening computes to

$$Pr[\mathcal{D}^*_{\alpha_{\mathrm{SA}}}] = 1 - \frac{\binom{n-m}{k}}{\binom{n}{k}}.$$

Using this approach we get a strictly better bound for our δ compared to using the straightforward amplification approach, i.e.,

$$Pr[\mathcal{D}^*_{\alpha_{\mathrm{SA}}}] < dPr[\mathcal{D}_{\alpha_{\mathrm{SA}}}].$$

The extent to which this is better varies and depends on the parameters n, k and m and is further elaborated on in the extended version [33].

D. System-Level Attacks and Adaptations

Next, we consider attacks that are not directly covered by our model and explore how the strong adversary we employ helps to deal with them. We then analyze the *entry guard* mechanism, a feature of the Tor protocol, and its influence on sender anonymity.

1) Traffic Analysis Attacks: Many of the known attacks on Tor nowadays depend on so called side-channel information, i.e. throughput and timing information an adversary might gather while watching traffic routed through the Tor network. Since the UC framework does not allow time-sensitive attacks, traffic analysis is outside of the scope of this work. However, due to the strong adversary we deploy, we can still cover all known attacks by making suitable assumptions. In the following we look at two well known traffic analysis attacks and how we can cover them in our model.

Traffic Correlation. These forms of traffic analysis attacks observe traffic going out from the sender and into the receiver and try to correlate them based on different features like volume, direction or inter-packet delay [38], [39]. We cover these attacks by assuming that the adversary knows which row of the input tables was being transmitted for each of the messages he intercepts. This enables him to find out who communicates with whom by simply compromising entry- and exit-node of the same circuit. This is made explicit in our extension of $\mathcal{F}_{\mathrm{OR}}$ for traffic analysis which can be found in the extended version [33].

Website Fingerprinting. Fingerprinting attacks try to classify user traffic based on a catalog of fingerprints derived for a large set of web pages beforehand and matching the observed traffic to those fingerprints [40], [41], [42]. This kind of attack can be modeled by assuming that it is enough for the adversary to compromise the entry node (i.e. we define a new distinguishing event $\mathcal{D}_{WF}$ that captures this) to find the recipient, as he will then be able to launch the fingerprinting attack. The δ in Theorem 19 then changes to

$$\delta = Pr[\mathcal{D}_{WF}] \cdot Pr[\mathcal{S}]$$

where $\mathcal{S}$ is the event that the website fingerprinting attack successfully classifies the traffic.

2) Entry Guards: Using the formulation for more than one challenge row, we can also motivate *entry guards* [43], [44], which are used in the current implementation of Tor. Entry guards are a small subset of the whole set of onion routers that are chosen by a user before the initiation of a Tor communication. They are then used as entry nodes for any subsequent communication. The advantage of this concept becomes apparent if we look at the following scenario:[3] Consider a single user u who communicates using Tor over a long period of time, initiating a total of d new sessions. Without entry guards, the probability that $\mathcal{A}$ de-anonymizes u is bounded by

$$1 - \frac{\binom{n-d}{k}}{\binom{n}{k}}$$

which converges to 1 the bigger d gets. If we do use a set of m entry guards on the other hand, the probability for de-anonymizing u will stay constant at

$$1 - \frac{\binom{n-m}{k}}{\binom{n}{k}}.$$

In order to prevent loss of performance, entry guards are also replaced at regular intervals. Let l be the maximum number of sessions possible per entry-guard-interval. The probability for de-anonymization can then be bounded by

$$1 - \frac{\binom{n-\lceil \frac{d}{l} \rceil m}{k}}{\binom{n}{k}}$$

[3] We consider the sender anonymity setting, i.e. we are only interested in entry nodes.

which is smaller than the original value, but still converges to 1 at some point. Note that these upper bounds only make sense if the sessions initiated per entry-guard-interval also use each entry-guard at least once. Dropping this assumption requires a more fine-grained analysis.

The problem with entry guards is the following: while the probability for de-anonymization is smaller, u will effectively stay de-anonymized as soon as $\mathcal{A}$ manages to find u's entry guards (for as long as these entry guards are used). Also, while the above value attains its minimum for $m = 1$, choosing a small value for m will realistically also incur loss in performance for the whole system. The exact analysis is unfortunately out-of-scope for our approach, but further elaboration on the parameters and their influence on anonymity and performance using simulation can be found in Elahi et al. [45].

E. Link-Corruption

So far we have only been concerned with an adversary $\mathcal{A}$ that compromises nodes in the onion routing network in order to learn about the transmitted messages. But our model also supports an adversary that compromises links between nodes and learns about messages transmitted through these links.

Thus, the event $\mathcal{D}_{\alpha_{\text{SA}}}$ alone is not enough to capture all bad events. For sender anonymity, we also lose if the adversary manages to compromise the link between the user and the entry node of the circuit used to transmit the challenge row. Let $\mathcal{L}_{\alpha_{\text{SA}}}$ be the event that this entry link is compromised and let q be the number of compromised links. Naturally, it is in the best interest of the adversary to not compromise links between user/server and already compromised nodes, as he will not learn anything new that way. Hence we have that

$$Pr[\mathcal{L}_{\alpha_{\text{SA}}}] \leq \frac{q}{n-k}$$

In order to extend our δ by the event $\mathcal{L}_{\alpha_{\text{SA}}}$, we can now consider the "bad event" $\mathcal{B}_{\alpha_{\text{SA}}}$ depending on $\mathcal{D}_{\alpha_{\text{SA}}}$:

$$\begin{aligned}
Pr[\mathcal{B}_{\alpha_{\text{SA}}}] &= \Pr[\mathcal{B}_{\alpha_{\text{SA}}}|\mathcal{D}_{\alpha_{\text{SA}}}] \cdot Pr[\mathcal{D}_{\alpha_{\text{SA}}}] \\
&\qquad + Pr[\mathcal{B}_{\alpha_{\text{SA}}}|\neg\mathcal{D}] \cdot Pr[\neg\mathcal{D}_{\alpha_{\text{SA}}}] \\
&= Pr[\mathcal{D}_{\alpha_{\text{SA}}}] + Pr[\mathcal{L}_{\alpha_{\text{SA}}}] \cdot Pr[\neg\mathcal{D}_{\alpha_{\text{SA}}}] \\
&\leq \frac{k}{n} + \frac{q}{n-k}\frac{n-k}{n} \\
&= \frac{k+q}{n}
\end{aligned}$$

For more than one challenge row this can be extended in a similar way as before, by just adjusting the event for successful link corruption. Let $\mathcal{L}^*_{\alpha_{\text{SA}}}$ be the event that in one of the challenge rows, an entry-link was successfully compromised. Doing a similar analysis as for the node corruption, we get the following upper bound, which is tight if the user for all challenge rows is the same.

$$Pr[\mathcal{L}^*_{\alpha_{\text{SA}}}] \leq 1 - (1 - \frac{q}{n-k})^d \tag{1}$$

The full derivation of Inequality 1 can be found in the extended version [33].

This concludes the formal analysis of the Tor network with the AnoA framework. We illustrated how AnoA can be used by using it on $\mathcal{F}_{\text{OR}}$. We showed that $\mathcal{F}_{\text{OR}}$ is $(0, \delta)$-α-IND-CDP for the different anonymity notion we defined in Section III and also explored further aspects of OR anonymity accessible through the AnoA framework. Still, we barely scratched the surface with our analysis and see many different directions for future work in the Tor analysis with AnoA. We further elaborate on these directions in Section VIII.

Note that, although we only considered the ideal functionality $\mathcal{F}_{\text{OR}}$ in our analysis, Theorem 17 allows us to lift our results to any (cryptographic) protocol that realizes $\mathcal{F}_{\text{OR}}$.

VII. Related Work

Pfitzmann and Hansen [32] 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 and definitions in our AnoA framework.

Our relaxation of differential privacy is not the first variation of differential privacy. Gehrke et al. recently introduced the stronger notion of zero-knowledge privacy [46] and the relaxed notion of crowd-blending privacy [47]. Similar to differential privacy, these notions are not well suited for the analysis of AC protocols. However, extending the crowd-blending privacy notion with corruptible distributed mechanisms and flexible adjacency functions would allow capturing the notion of k-anonymity for AC protocols. We could imagine applying the resulting concept to Mixnets, in which each mix waits for a certain amount of time: if at least k messages arrived, these messages are then processed, otherwise they are discarded; however, discarding messages in such a way may not be acceptable in a real world application.

Efforts to formally analyze anonymity properties have already been made using communicating sequential processes (CSP) [48], epistemic logic [49], [7], Kripke structures [19], and probabilistic automata [13]. However, these formalisms have only being applied to simple protocols such DC-net. Since it's not clear if these frameworks can capture an adversary with auxiliary information, it seems difficult to model complex protocols such as onion routing and its traffic analysis attacks. It still presents an interesting challenge to relate the probabilistic notions among those mentioned above (e.g. [7], [13]) to our anonymity framework.

There have been analyses which focus on a particular AC protocol, such as [4], [9], [8], [12] for Mixnet, [50], [13] for DC-net, [3], [5] for Crowds, and [2], [6], [10], [11], [14] 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. (We refer the readers to [14, Sec. 5] for a detailed survey.) The most recent result [14] among these by Feigenbaum, Johnson and Syverson models the OR protocol in a simplified black-box abstraction, and studies a notion of relationship anonymity notion which is slightly different from ours: here the adversary wishes to identify the destination of a user's message. As we discussed in Section IV-C, this relationship anonymity notion is slightly weaker than ours. Moreover, their model is not flexible enough to extend to other system-level scenarios such fingerprinting attacks [40], [41], [42].

Hevia and Micciancio [51] introduce an indistinguishability based framework for the analysis of AC protocols. While they take a similar approach as in AnoA, there are some notable differences: The first difference is that their anonymity definition does not consider compromised parties; as a consequence, they only define qualitative anonymity guarantees. While the authors discuss corruption as a possible extension, for most real world AC protocols they would have to adjust their notion to a quantitative anonymity notion as in AnoA. The second difference is the strength of the adversary: we consider a stronger adversary which determine the order in which messages are sent through the network, whereas Hevia and Micciancio only allow the attacker to specify which party sends which messages to whom.

VIII. Conclusion and Future Directions

In this paper we have presented our generic framework AnoA. We have defined new, strong variants of anonymity properties like sender anonymity, sender unlinkability and relationship anonymity based on a novel relaxation of computational differential privacy, and presented how to concisely formulate them in AnoA. We have shown that our definitions of the anonymity guarantees accurately model prominent notions in the literature. We have also applied AnoA to the UC framework and shown that the results shown for ideal functionalities carry over to their secure cryptographic protocols.

Additionally, we have conducted an extensive analysis of the Tor network. We have validated the inherent imperfection of the current Tor standard in the presence of a significant fraction of compromised nodes, and we have given quantitative measures of the different forms of anonymity against passive adversaries that statically corrupt nodes. Naturally, the next step will be to investigate adaptively corrupting adversaries and active attacks on Tor such as selective DoS attacks [52]. We also plan to analyze the influence of Tor's node selection policies [53] and of *a priori* probability distributions over the users [14] on Tor's anonymity properties. Moreover, we will apply AnoA to other AC protocols such as Mixnets [17] and the DISSENT system [54].

On the framework level we will investigate other anonymity notions such as unobservability and undetectability [32], and their relation to the notions we already defined in this paper.

Acknowledgment

We thank Aaron Johnson and the anonymous reviewers for their useful suggestions. This work was partially supported by the German Universities Excellence Initiative, the ERC Grant End-2-End Security, and the Center for IT-Security, Privacy and Accountability (CISPA).

References

[1] "The Tor Project," https://www.torproject.org/, 2003, accessed Feb 2013.

[2] 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.

[3] 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.

[4] 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.

[5] V. Shmatikov, "Probabilistic Analysis of an Anonymity System," *Journal of Computer Security*, vol. 12, no. 3-4, pp. 355–377, 2004.

[6] S. Mauw, J. Verschuren, and E. de Vink, "A Formalization of Anonymity and Onion Routing," in *Proc. 9th European Symposium on Research in Computer Security (ESORICS)*, 2004, pp. 109–124.

[7] J. Y. Halpern and K. R. O'Neill, "Anonymity and Information Hiding in Multiagent Systems," *Journal of Computer Security*, vol. 13, no. 3, pp. 483–512, 2005.

[8] 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.

[9] C. Díaz, "Anonymity Metrics Revisited," in *Anonymous Communication and its Applications*, 2006.

[10] 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.

[11] ——, "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.

[12] 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.

[13] E. Andrés, Miguel, C. Palamidessi, A. Sokolova, and P. Van Rossum, "Information Hiding in Probabilistic Concurrent System," *Journal of Theoretical Computer Science (TCS)*, vol. 412, no. 28, pp. 3072–3089, 2011.

[14] J. Feigenbaum, A. Johnson, and P. F. Syverson, "Probabilistic Analysis of Onion Routing in a Black-Box Model," *ACM Transactions on Information and System Security (TISSEC)*, vol. 15, no. 3, p. 14, 2012.

[15] D. Chaum, "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability," *J. Cryptology*, vol. 1, no. 1, pp. 65–75, 1988.

[16] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web Transactions," *ACM Transactions on Information and System Security (TISSEC)*, vol. 1, no. 1, pp. 66–92, 1998.

[17] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, vol. 4, no. 2, pp. 84–88, 1981.

[18] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous Connections and Onion Routing," *IEEE J-SAC*, vol. 16, no. 4, pp. 482–494, 1998.

[19] D. Hughes and V. Shmatikov, "Information Hiding, Anonymity and Privacy: a Modular Approach," *Journal of Computer Security*, vol. 12, no. 1, pp. 3–36, 2004.

[20] C. Dwork, "Differential Privacy," in *ICALP (2)*, 2006, pp. 1–12.

[21] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *Proc. 10th Theory of Cryptography Conference (TCC)*, 2006, pp. 265–284.

[22] I. Mironov, O. Pandey, O. Reingold, and S. P. Vadhan, "Computational Differential Privacy," in *Advances in Cryptology — CRYPTO*, vol. 5677, 2009, pp. 126–142.

[23] 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.

[24] R. Küsters and M. Tuengerthal, "The IITM Model: a Simple and Expressive Model for Universal Composability," *IACR Cryptology ePrint Archive*, vol. 2013, p. 25, 2013.

[25] M. Backes, B. Pfitzmann, and M. Waidner, "The Reactive Simulatability (RSIM) Framework for Asynchronous Systems," *Information and Computation*, vol. 205, no. 12, pp. 1685–1720, 2007.

[26] D. Wikström, "A Universally Composable Mix-Net," in *Proc. of the 1st Theory of Cryptography Conference (TCC)*, 2004, pp. 317–335.

[27] J. Camenisch and A. Lysyanskaya, "A Formal Treatment of Onion Routing," in *Advances in Cryptology — CRYPTO*, 2005, pp. 169–187.

[28] G. Danezis and I. Goldberg, "Sphinx: A Compact and Provably Secure Mix Format," in *Proc. 30th IEEE Symposium on Security and Privacy*, 2009, pp. 269–282.

[29] A. Kate and I. Goldberg, "Using Sphinx to Improve Onion Routing Circuit Construction," in *Proc. 14th Conference on Financial Cryptography and Data Security (FC)*, 2010, pp. 359–366.

[30] 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.

[31] C. Díaz and B. Preneel, "Reasoning About the Anonymity Provided by Pool Mixes That Generate Dummy Traffic," in *Information Hiding*, 2004, pp. 309–325.

[32] 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, Aug. 2010, v0.34.

[33] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi, "AnoA: A Framework For Analyzing Anonymous Communication Protocols — Unified Definitions and Analyses of Anonymity Properties," available at http://www.infsec.cs.uni-saarland.de/~meiser/paper/anoa.html.

[34] 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.

[35] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," in *Proc. 13th USENIX Security Symposium (USENIX)*, 2004, pp. 303–320.

[36] "Tor Metrics Portal," https://metrics.torproject.org/, accessed Feb 2013.

[37] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our Data, Ourselves: Privacy Via Distributed Noise Generation," in *Advances in Cryptology — EUROCRYPT*, 2006, pp. 486–503.

[38] G. O'Gorman and S. Blott, "Improving Stream Correlation Attacks on Anonymous Networks," in *Proceedings of the 2009 ACM Symposium on Applied Computing (SAC)*, 2009, pp. 2024–2028.

[39] X. Wang, D. S. Reeves, and S. F. Wu, "Inter-Packet Delay Based Correlation for Tracing Encrypted Connections through Stepping Stones," in *Proc. 7th European Symposium on Research in Computer Security (ESORICS)*, 2002, pp. 244–263.

[40] 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.

[41] 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.

[42] 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 *Proc. 33th IEEE Symposium on Security and Privacy*, 2012, pp. 332–346.

[43] M. Wright, M. Adler, B. N. Levine, and C. Shields, "Defending Anonymous Communication Against Passive Logging Attacks," in *Proc. 24th IEEE Symposium on Security and Privacy*, 2003, pp. 28–43.

[44] L. Øverlier and P. F. Syverson, "Locating Hidden Servers," in *Proc. 27th IEEE Symposium on Security and Privacy*, 2006, pp. 100–114.

[45] T. Elahi, K. S. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg, "Changing of the Guards: A Framework for Understanding and Improving Entry Guard Selection in Tor," in *Proc. 11th ACM Workshop on Privacy in the Electronic Society (WPES)*, 2012, pp. 43–54.

[46] J. Gehrke, E. Lui, and R. Pass, "Towards Privacy for Social Networks: A Zero-Knowledge Based Definition of Privacy," in *Proc. 8th Theory of Cryptography Conference (TCC)*, 2011, pp. 432–449.

[47] J. Gehrke, M. Hay, E. Lui, and R. Pass, "Crowd-Blending Privacy," in *Advances in Cryptology — CRYPTO*, 2012, pp. 479–496.

[48] S. Schneider and A. Sidiropoulos, "CSP and Anonymity," in *Proc. 4th European Symposium on Research in Computer Security (ESORICS)*, 1996, pp. 198–218.

[49] P. F. Syverson and S. G. Stubblebine, "Group Principals and the Formalization of Anonymity," in *World Congress on Formal Methods*, 1999, pp. 814–833.

[50] M. Bhargava and C. Palamidessi, "Probabilistic Anonymity," in *CONCUR*, 2005, pp. 171–185.

[51] A. Hevia and D. Micciancio, "An Indistinguishability-Based Characterization of Anonymous Channels," in *Proc. 8th Privacy Enhancing Technologies Symposium (PETS)*, 2008, pp. 24–43.

[52] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, "Denial of Service or Denial of Security?" in *Proc. 14th ACM Conference on Computer and Communications Security (CCS)*, 2007, pp. 92–102.

[53] R. Dingledine and S. Murdoch, "Performance Improvements on Tor or, Why Tor is slow and what we're going to do about it," *Online: http://www. torproject. org/press/presskit/2009-03-11-performance. pdf*, 2009.

[54] H. Corrigan-Gibbs and B. Ford, "Dissent: Accountable Anonymous Group Messaging," in *Proc. 17th ACM Conference on Computer and Communication Security (CCS)*, 2010, pp. 340–350.