AnoA: A Framework For Analyzing Anonymous Communication Protocols*

Anonymity meets differential privacy

Michael Backes[1,3], Aniket Kate[2],
Praveen Manoharan[3], Sebastian Meiser[3], and Esfandiar Mohammadi[3]

[1] MPI-SWS
[2] MMCI, Saarland University
[3] Saarland University
{backes,manoharan,meiser,mohammadi}@cs.uni-saarland.de
aniket@mmci.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.

1 Introduction

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 [40].

There has been a substantial amount of previous work [38, 13, 35, 36, 28, 24, 37, 12, 18, 19, 23, 2, 20] on analyzing the anonymity provided by various AC protocols such as dining cryptographers network (DC-net) [10], Crowds [33], mix network (Mixnet) [9], and onion routing (e.g., Tor) [32]. However, most of the previous works only consider a single anonymity property for a particular AC protocol under a specific adversary scenario.

* This work appeared at IEEE CSF 2013.

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.

1.1 Contributions

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. [15, 16]. 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 recipient. We overcome this contradiction by generalizing the adjacency of settings between which an adversary has to distinguish by introducing an explicit *adjacency function* α.

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.

In the extended version [3] we also compare our formalizations of the anonymity notions. Additionally we apply our framework to the most successful AC protocol—Tor. We consider known system-level attacks, such as website fingerprinting and traffic correlation and a known countermeasure for Tor's high sensitivity to compromised nodes: the entry guards mechanism. 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.

2 Notation

We 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 $i \xleftarrow{R} I$ 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 $\Pr[b = 1 : b \xleftarrow{R} \{0,1\}]$ 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.

3 The AnoA Framework

3.1 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. 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 multi-party protocols in classical simulation-based composability frameworks, such as the UC [8], the IITM [27] or the RSIM [4] 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.

3.2 Generalized Computational Differential Privacy

For privacy preserving computations the notion of *differential privacy* (DP) [15, 16] 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) [29]. 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 1: 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 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, Ch chooses D'_b, depending on its input bit b^*, and successively feeds one row after the other to the pro-

tocol $\mathcal{P}$.[4] 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 party u. In detail, upon a message $(\mathsf{input}, D_0, D_1)$ sent by $\mathcal{A}$, $\textsc{Ch}(\mathcal{P}, \alpha, b^*)$ computes $(D'_0, D'_1) \leftarrow \alpha(D_0, D_1)$. If $(D'_0, D'_1) \neq \perp$, $\textsc{Ch}$ 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.

Our definition depends on two parameters: ϵ and δ. As in the definition of differential privacy, ϵ quantifies the degree of anonymity (see Example 3). The anonymity of commonly employed AC protocols also break down if certain distinguishing events happen, e.g., when an entry guard of a Tor user is compromised. Similar to CDP, the probability that such a distinguishing event happens is quantified by the parameter δ. However, in contrast to CDP, this δ is typically non-negligible and depends on the degree of corruption in the AC network. As a next step, we formally define (ε, δ)-α-IND-CDP.

Definition 2 ((ε, δ)-α-IND-CDP). *Let* $\textsc{Ch}$ *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* $\mathcal{A}$*:*

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

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 [18, 20].

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* $\mathsf{CR}(D)$*.*

3.3 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.

[4] In contrast to IND-CDP, we only consider PPT-computable tables.

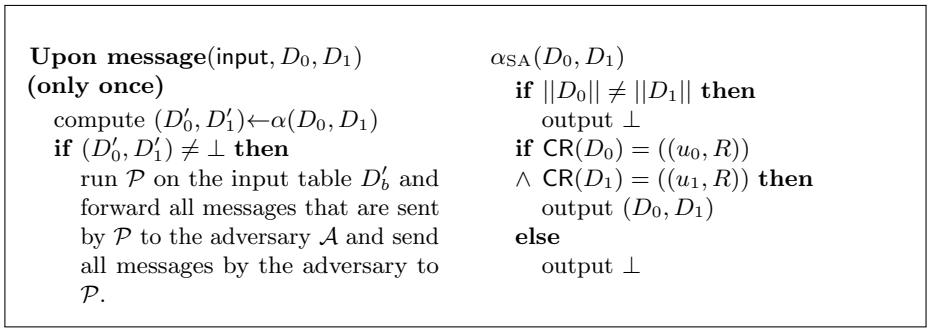

Fig. 1. The challenger $\text{CH}(\mathcal{P}, \alpha, b)$ for the adjacency function α and the adjacency function α_{SA} for sender anonymity.

Third, we define relationship anonymity, which states that an adversary (that potentially controls some protocol parties) cannot relate sender and recipient in a communication.

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 practical AC protocols is infeasible.

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 4 we exactly quantify the gap between our notion and the notion from the literature.

We formalize our notion of sender anonymity with the definition of an adjacency function α_{SA} as depicted in Figure 1. 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 1.*

Example 2: Sender anonymity. The adversary $\mathcal{A}$ decides that he wants to use users Alice and Bob in the 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_{SA} 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 $\perp$. $\diamond$

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

The value of ε. In the extended version of this paper [3] we analyze the widely used AC protocol Tor. We show that if every node is uniformly selected then Tor satisfies sender anonymity with $\varepsilon = 0$. If the nodes are selected using preferences, e.g., in order to improve throughput and latency, ε and δ may increase.[5]

Recall that the value δ describes the probability of a distinguishing event, and if this distinguishing event occurs, anonymity is broken. In the sender anonymity game for Tor this event occurs if the entry guard of the user's circuit is compromised. If a user has a preference for the first node, the adversary can compromise the most likely node. Thus, a preference for the first node in a circuit increases the probability for the distinguishing event (δ). However, if there is a preference for the second node in a circuit, corrupting this node does not lead to the distinguishing event but can still increase the adversary's success probability by increasing ε. Consider the following example.

Example 3: The value of ε. *Assume that the probability that Alice chooses a specific node N as second node is $\frac{1}{40}$ and the probability that Bob uses N as second node is $\frac{3}{40}$. Further assume that for all other nodes and users the probabilities are uniformly distributed. Suppose the adversary $\mathcal{A}$ corrupts N. If $\mathcal{A}$ observes communication over the node N, the probability that this communication originates from Bob is 3 times the probability that it originates from Alice. Thus, with such preferences Tor only satisfies sender anonymity with $\varepsilon = \ln 3$.* $\diamond$

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 [31]. 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 2, 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 $\text{CH}(\mathcal{P}, \alpha_{\text{UL}}, 0)$ and $\text{CH}(\mathcal{P}, \alpha_{\text{UL}}, 1)$. 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 2.*

Example 4: 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*

[5] Previous work discusses the influence of node selection preferences on Tor's anonymity guarantees, e.g., [1].

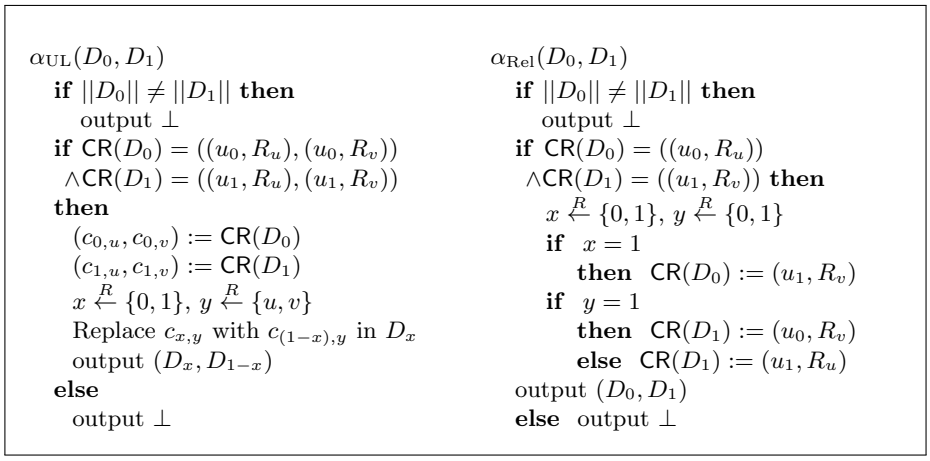

Fig. 2. The adjacency function α_{Rel} for relationship anonymity. and the adjacency function α_{UL} for sender unlinkability.

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. ◇

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 [31]. 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 $\textsc{Ch}(\mathcal{P}, \alpha_{\mathrm{Rel}}, 0)$ and $\textsc{Ch}(\mathcal{P}, \alpha_{\mathrm{Rel}}, 1)$.

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 2.*

Example 5: 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 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). $\diamond$

4 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 [31].

4.1 Sender Anonymity

The notion of sender anonymity is introduced in [31] 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 $\mathcal{A}$*

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

where the challenger SAC$_\text{H}$ *as defined as in Figure 3.*

Note that SAC$_\text{H}$ slightly differs from the challenger C$_\text{H}(\mathcal{P}, \alpha, b)$ 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 [3].

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.*

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}.*

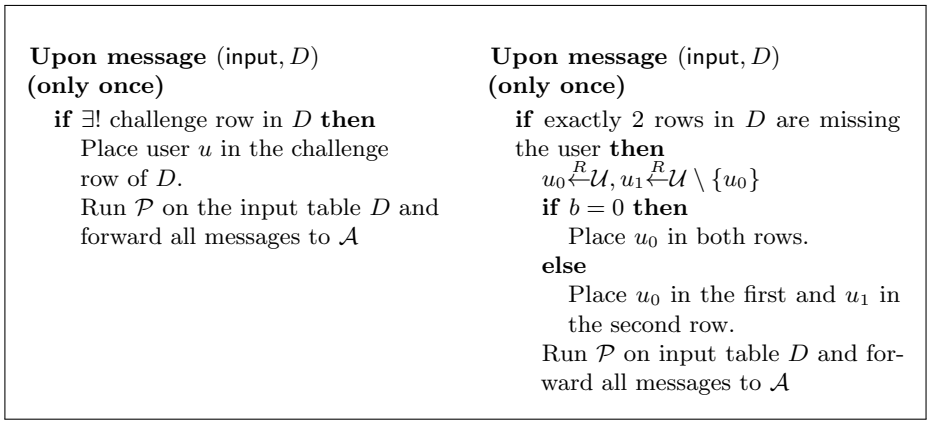

Fig. 3. The challenger $\text{ULC}_\text{H}(\mathcal{P}, b)$ and the challenger $\text{SAC}_\text{H}(\mathcal{P}, u)$

4.2 Unlinkability

The notion of unlinkability is defined in [31] 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 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 $\mathcal{A}$*

$$\left| Pr\left[b = 0 : b \leftarrow \mathcal{A}^{\text{ULC}_\text{H}(\mathcal{P},0)}\right] - Pr\left[b = 0 : b \leftarrow \mathcal{A}^{\text{ULC}_\text{H}(\mathcal{P},1)}\right] \right| \leq \delta$$

where the challenger ULCH *is as defined in Figure 3.*

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.*

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}.*

4.3 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 [37] define relationship anonymity as 'hiding the fact that party A is communicating with party B'. Feigenbaum et al. [19] 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 [31]: 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 anonymity because the adversary needs to uncover both sender and recipient in order to break anonymity.

5 Related Work

Pfitzmann and Hansen [31] 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 (DP). Gehrke et al. recently introduced the stronger notion of zero-knowledge privacy [22] and the relaxed notion of crowd-blending privacy [21]. Similar to DP, these notions are not 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 arrive, they are 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) [34], epistemic logic [39, 24], Kripke structures [26], and probabilistic automata [2]. 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. [24, 2]) to our anonymity framework.

There have been analyses which focus on a particular AC protocol, such as [35, 12, 37, 23] for Mixnet, [5, 2] for DC-net, [13, 36] for Crowds, and [38, 28, 18, 19, 20] 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. The most recent result [20] 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: they require the adversary to identify the destination of a user's message. As discussed in Section 4.3, this formalization of relationship anonymity is weaker than ours. Moreover, it is not clear how to extend their model to other system-level scenarios such fingerprinting attacks [30, 7, 17].

Hevia and Micciancio [25] 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: our adversary can 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.

6 Future Directions

In the extended version [3], we conduct a thorough analysis of Tor against passive attackers and prove sender anonymity, sender unlinkability and relationship anonymity for $\varepsilon = 0$. In our analysis of Tor we did not consider the impact of preferences. If certain nodes are more likely for a given user (e.g. for efficiency reasons), anonymity can (and will) decrease. As illustrated in Example 3, when analyzing Tor with preferences, the value for ε may be larger than zero.

The next step will be to investigate adaptively corrupting adversaries and active attacks on Tor such as selective DoS attacks [6]. We also plan to analyze the influence of Tor's node selection policies [14] and of *a priori* probability distributions over the users [20] on Tor's anonymity properties. Moreover, we

will apply AnoA to other AC protocols such as Mixnets [9] and the DISSENT system [11].

On the framework level we will investigate other anonymity notions such as unobservability and undetectability [31], 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] 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)*, pages 476–490. IEEE Computer Society, 2012.

[2] E. Andrés, Miguel, C. Palamidessi, A. Sokolova, and P. Van Rossum. Information Hiding in Probabilistic Concurrent System. *Journal of Theoretical Computer Science (TCS)*, 412(28):3072–3089, 2011.

[3] 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`.

[4] M. Backes, B. Pfitzmann, and M. Waidner. The Reactive Simulatability (RSIM) Framework for Asynchronous Systems. *Information and Computation*, 205(12):1685–1720, 2007.

[5] M. Bhargava and C. Palamidessi. Probabilistic Anonymity. In *CONCUR*, pages 171–185, 2005.

[6] 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)*, pages 92–102, 2007.

[7] 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)*, pages 605–616, 2012.

[8] R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 136–145, 2001.

[9] D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 4(2):84–88, 1981.

[10] D. Chaum. The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. *J. Cryptology*, 1(1):65–75, 1988.

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

[12] C. Díaz. Anonymity Metrics Revisited. In *Anonymous Communication and its Applications*, 2006.

[13] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards Measuring Anonymity. In *Proc. 2nd Workshop on Privacy Enhancing Technologies (PET)*, pages 54–68, 2002.

[14] 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.
[15] C. Dwork. Differential Privacy. In *ICALP (2)*, pages 1–12, 2006.
[16] 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)*, pages 265–284, 2006.
[17] 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*, pages 332–346, 2012.
[18] 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)*, pages 57–71, 2007.
[19] 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)*, pages 1–10, 2007.
[20] 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)*, 15(3):14, 2012.
[21] J. Gehrke, M. Hay, E. Lui, and R. Pass. Crowd-Blending Privacy. In *Advances in Cryptology — CRYPTO*, pages 479–496, 2012.
[22] 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)*, pages 432–449, 2011.
[23] 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)*, pages 111–116, 2008.
[24] J. Y. Halpern and K. R. O'Neill. Anonymity and Information Hiding in Multiagent Systems. *Journal of Computer Security*, 13(3):483–512, 2005.
[25] A. Hevia and D. Micciancio. An Indistinguishability-Based Characterization of Anonymous Channels. In *Proc. 8th Privacy Enhancing Technologies Symposium (PETS)*, pages 24–43, 2008.
[26] D. Hughes and V. Shmatikov. Information Hiding, Anonymity and Privacy: a Modular Approach. *Journal of Computer Security*, 12(1):3–36, 2004.
[27] R. Küsters and M. Tuengerthal. The IITM Model: a Simple and Expressive Model for Universal Composability. *IACR Cryptology ePrint Archive*, 2013:25, 2013.
[28] 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)*, pages 109–124, 2004.
[29] I. Mironov, O. Pandey, O. Reingold, and S. P. Vadhan. Computational Differential Privacy. In *Advances in Cryptology — CRYPTO*, volume 5677, pages 126–142, 2009.
[30] 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)*, pages 103–114, 2011.
[31] 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.
[32] M. Reed, P. Syverson, and D. Goldschlag. Anonymous Connections and Onion Routing. *IEEE J-SAC*, 16(4):482–494, 1998.

[33] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security (TISSEC)*, 1(1):66–92, 1998.

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

[35] A. Serjantov and G. Danezis. Towards an Information Theoretic Metric for Anonymity. In *Proc. 2nd Workshop on Privacy Enhancing Technologies (PET)*, pages 41–53, 2002.

[36] V. Shmatikov. Probabilistic Analysis of an Anonymity System. *Journal of Computer Security*, 12(3-4):355–377, 2004.

[37] V. Shmatikov and M.-H. Wang. Measuring Relationship Anonymity in Mix Networks. In *Proc. 7th ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 59–62, 2006.

[38] 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)*, pages 96–114, 2000.

[39] P. F. Syverson and S. G. Stubblebine. Group Principals and the Formalization of Anonymity. In *World Congress on Formal Methods*, pages 814–833, 1999.

[40] The Tor Project. `https://www.torproject.org/`, 2003. Accessed Feb 2013.