On the Limits of Provable Anonymity

Nethanel Gelernter
Department of Computer Science
Bar Ilan University
nethanel.gelernter@gmail.com

Amir Herzberg
Department of Computer Science
Bar Ilan University
amir.herzberg@gmail.com

ABSTRACT

We study *provably secure anonymity*, focusing on *ultimate anonymity* - strongest-possible anonymity requirements and adversaries. We begin with rigorous definition of anonymity against wide range of computationally-bounded attackers, including eavesdroppers, malicious peers, malicious destinations, and their combinations. Following [15], our definition is generic, and captures different notions of anonymity (e.g., unobservability and sender anonymity).

We then study the *feasibility of ultimate anonymity*. We show there is a protocol satisfying this requirement, but with absurd (although polynomial) inefficiency and overhead. We show that such inefficiency and overhead is unavoidable for 'ultimate anonymity'. We then present a slightly-relaxed requirement and present feasible protocols for it.

1. INTRODUCTION

Anonymous communication is an important goal, and is also interesting and challenging. Since the publication of the first, seminal paper by Chaum [7], there has been a large research effort by cryptography and security researchers to study anonymity and develop solutions, resulting in numerous publications and several systems.

Research of anonymous communication is challenging; indeed, it is not even easy to agree on good definitions. Much of the research uses entropy-based definitions, e.g., the probability of identifying the sender must be lower than some threshold. Syverson discuss in depth the limitations of this definitional approach [24], and in particular, the fact that it fails to capture the capabilities and limitations of the attacker.

Our goal is to study rigorous definitions, capturing the *strongest possible and feasible definitions of anonymous communication.* Following the approach of [24], we focus on well-defined adversary capabilities, and present a rigorous, indistinguishability-based definition, considering the strongest-possible adversaries and the strongest anonymity requirements

We note that such rigorous study of anonymous communication, may necessarily involve complex definitions; this probably explains the fact that with so much research on anonymous communication, not many works use rigorous models. Specifically, our work extends the definitions of Hevia and Micciancio [15], which are based on an indistiguishability experiment: the attacker chooses two scenarios and the experiment simulates one of them; the attacker should distinguish which scenario was simulated.

In [15], the adversary was limited, and in particular was only 'eavesdropper' - it could not control any participant, in particular, not the destination. These limitations are very significant; in fact, most of the efforts to develop and research anonymous communication, in particular deployed anonymity systems, focused on anonymity against a (malicious) destination; malicious peers are also often considered. We extend [15] to deal with such realistic threats.

Our extended definitions allow adversary to control active, malicious peers and destination. This requires us to define precise model and experiments. These are (even) more complex that these of [15]; however, this complexity may be unavoidable when trying to rigorously study anonymity. (One obvious challenge for future research is to present simple models and definitions.)

Dealing with a malicious *destination* is esp. challenging. Indeed, many of the anonymity properties considered in the *common terminology* of Pfitzmann and Hansen [17–19], e.g., unobservability, are trivially inapplicable against a malicious destination (which can observe received traffic). We conclude, that the 'ultimate' anonymity, requires the strongest properties achievable against malicious destination, and in addition, the strongest properties achievable assuming a benign destination.

Another challenge we had to deal with, is that a strong adversary should be allowed to be *adaptive*. As with many cryptographic primitives, there is a significant difference between adaptive and non-adaptive adversaries (for example CCA1 and CCA2 encryption schemes [2]), and between passive and active attackers (for example security against semi honest or malicious adversaries in multi party computation protocols [13]). To deal with adaptive and active attackers, we had to define a simulation model for the tested protocols. This challenge was not relevant or addressed in previous works [15].

Using our definitions and model, it is possible to formally prove different anonymity notions with respect to different attacker capabilities. We define the capability of the attacker by the protocol's participants it controls, and the participants to whom it can eavesdrop. Protocols can have different anonymity notions against different attackers with different capabilities.

Our definitions and adversary model are bit complex, but we believe this is the necessary cost of giving rigorous formal definition for different anonymity notions, against wide range of attackers.

1.1 Contributions

Our main contribution is in presenting rigorous, indistinguishability based definitions for anonymous communication protocols, whose anonymity is assured even against strong, malicious, adaptive attackers, which may control nodes, possibly including the destination. Previous rigorous definitions [15] were limited to eavesdropping attackers, not even ensuring anonymity against the destination; therefore, this is significant, critical extension.

We actually explore two variants of this definition. The stronger requirements essentially formalizes the strongest anonymity considered in the literature, e.g., in the common terminology [17–19]. We show it is possible to achieve this variant, albeit, with an inefficient protocol (more a 'proof of feasibility' than a real protocol). We further show, that this inefficiency is unavoidable, i.e., we prove that *any* protocol meeting this variant of the definition, would be very inefficient. This motivates slightly relaxing the anonymity requirements, as we do in our second definition. Indeed, we show that this slightly-relaxed definition can be satisfied, with reasonable efficient protocols. For example, the classical DC-net protocol [8] that fails to satisfy the stronger requirement, does satisfy this slightly weaker requirement. In the full version, we also present improved protocols, which ensure this anonymity property even against multiple malicious nodes.

Organization

In Section 2, we formally define the adversary model, and present our experiment based definition. In Section 3, we extend the definition to consider also malicious destination. In Section 4, we discuss the feasibility of the definitions of the previous section against strong attackers. In Section 5 we present slightly relaxed definition for some of the anonymity notions against malicious destination, and in the last section we conclude and discuss future directions.

1.2 Related Works

There is a huge body of research in theory and practice of anonymous communication, beginning with Chaum's paper [7]; see, e.g., a survey of known protocols in [21]. Even just focusing on the closely related works, focusing on rigorous definitions, would far exceed the length limitations, and is delegated to the full version. A good overview of related rigorous works appears in [15], where Hevia and Micciancio presented rigorous, indistinguishability-based definitions to most anonymity notions, limited to passive, non-destination adversaries. Our work extends [15] to deal with strong, active, malicious attackers, including destination.

Few recent works extend [15] in different ways, e.g., applying the UC framework [6] for anonymous communication [25], and further studying relations among the notions [5, 16]. However, these works do not address our goals of studying the strongest anonymity notions (against strongest adversaries).

The latest version of the common terminology [18] contains comparison between the terminology to the anonymity notions in [15].

A framework for formalizing and comparing identity-related properties is offered in [26], however, differently from our approach, they ignore the confidentially of the messages content, and therefore cannot capture many of the anonymity notions our definition captures (e.g., see section 6.2 there).

Other works, offer formal analysis that is limited to specific protocols. In [1] and [10] the Onion-Routing (OR) [20] protocol is discussed; the authors present definitions for OR in the UC framework [6]. In [1] the authors further discuss the security properties required for OR cryptographic primitives, needed to achieve provable security.

2. DEFINITIONS

Following Hevia and Micciancio [15], we offer definition that is based on an experiment that simulates protocol run over some network. We let the adversary choose between two scenarios. The "relation" between the scenarios is restricted by the anonymity notion **N** that is tested in the experiment and by the capability of the attacker. The adversary controls the scenarios, by controlling the application level of all the protocol participants: who sends what to whom in both scenarios. This is done by periodically choosing two matrices of messages, $M^{(0)}$ and $M^{(1)}$, one for each scenario. We define two experiments. The first simulates the protocols by the $M^{(0)}$ matrices, and the second by $M^{(1)}$ matrices. The adversary, that gets information about the protocol simulation by its capability (for example: global eavesdropper gets all the traffic), has to distinguish between the experiments, by guessing which world was simulated.

2.1 Network model, adversary and peers

Since our goal is to study anonymity against adaptive and active attackers, we need a rigorous communication and execution model. In this work, we adopt the simplest model: fully synchronous ('rounds/iterations') communication with instantaneous computation, allowing direct communication between every two participants (clique).

Peers. We let the adversary control the 'application layer' of all peers, i.e., deliver requests to the protocol layer, to send messages to particular destination(s). In the protocol layer, the honest peers follow the protocol and are simulated by the experiment, while the attacker controls the 'malicious peers'.

Different peers can have different roles in the protocol; for example, protocols that use mixes [7, 22] or routers [9] to assist anonymous communication by other peers, often have two types of peers: client and mix (or router). The roles of the participants are determined by the protocol.

Adversary. The attacker controls the application layer of all the peers. Namely, the attacker chooses, for every peer and at every round, a matrix of messages (from each peer, and to each peer). The anonymity requirements are defined by an indistinguishability game, where the attacker selects two sets of matrices (for each round) and the game selects one of them, and the adversary tries to detect which of the two sets was selected. Different anonymity notions, are represented by different restrictions on the matrices. In peers that it controls, the attacker can also deviate arbitrarily from the protocol (i.e., act in a malicious/byzantine manner).

The power to select the entire sequence of messages to be sent (the matrices) might seem excessive. However, this follows the same 'conservative' approach applied in experiments of cryptographic primitives such as encryption [3] [2]. As mentioned in [15], in reality, the attacker might have some influence on the application level of its victims. We conservatively give the attacker the whole control, as we cannot predict the attacker's influence about the application in different real scenarios.

2.2 Experiment: parameters, notations and security notions

Notations. We use the following common cryptographic and mathematical notations: $\mathcal{PPT}$ is the set of probabilistic polynomial time algorithms. For $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, 2, ..., n\}$. We use $\mathcal{P}(S)$ to denote the power set of set S. Consider two sets, S_1 and S_2, then $S_1 \in S_2{}^*$ if and only for every $s \in S_1$, $s \in S_2$.

We use $V = \{0, 1\}^l$ to denote the messages space. A collection of messages between n parties is represented by an $n \times n$ matrix, $M = (m_{i,j})_{i,j \in [n]}$. Each element $m_{i,j} \in V^*$ is the multiset of messages from the i-th party to the j-th party [1].

2.2.1 The experiment parameters: $\pi, n, \mathcal{A}$ and *Cap*.

The first two parameters of the experiment, π and n, represent the protocol. π is a $\mathcal{PPT}$ algorithm that represents the tested protocol, and n is the number of participants in the protocol simulation, $n < l(k)$ when $l(\cdot)$ is some polynomial, and k is the security parameter of the experiment. To initialize the parties, e.g., establish shared (or public/private) keys, we use π's *setup* method, which receives the number of participants n and the identity i of a specific participant as parameter, and outputs the initial state of i (denoted by $STATE_i$); this follows the 'common reference string' model. In practice, ths simply means that we assume the parties have appropriate keys (shared or public/private). The π's *simulate* method receives the current state of a participant, together with its incoming traffic and new messages from the application layer, and returns its next state and its outgoing traffic.

The last two experiment's parameters, $\mathcal{A}$ and *Cap*, define the attacker. $\mathcal{A}$ is the attacker $\mathcal{PPT}$ algorithm. The *Cap* parameter defines the attacker capabilities, and consists of two sub-parameters, i.e., $Cap \in \mathcal{P}([n])^2$, which we denote $Cap = (\overline{H}, EV)$. The $\overline{H}$ parameter specifies the machines controlled by adversary $\mathcal{A}$, and the EV parameter identifies machines to which the attacker can eavesdrop (e.g., all machines, for a global eavesdropper). To refer to a specific parameter of *Cap* we use the notation $Cap[\overline{H}]$ and $Cap[EV]$ respectively. An attacker with capability $Cap = (\overline{H}, EV)$, controls the machines with indexes in $Cap[\overline{H}]$ and eavesdrops the traffic of the machines with indexes in $Cap[EV]$.

In the next section, we extend the capability to deal also with malicious destination, by adding to *Cap* another bit, $Cap[MD]$; the definition of this section is the same as that definition, using $MD = 0$.

2.2.2 Security notions

Following [15], the unprotected data (the attacker assumed knowledge), is defined by the functions $f_\cup$, f_Σ and $f_\#$ that map matrices from $\mathcal{M}_{n\times n}(V^*)$ into V^*, $\mathbb{N}^n$ and $\mathbb{N}$ respectively:

$$f_\cup(M) \overset{def}{=} (\uplus_{j\in[n]} m_{i,j})_{i\in[n]}$$

$$f_\Sigma(M) \overset{def}{=} (\sum_{j\in[n]} |m_{i,j}|)_{i\in[n]}$$

$$f_\#(M) \overset{def}{=} \sum_{i,j\in[n]} |m_{i,j}|$$

Additionally, we define $f^T_\cup(M) \overset{def}{=} f_\cup(M^T)$ and similarly $f^T_\Sigma(M) \overset{def}{=} f_\Sigma(M^T)$. For a given function $f \in \{f_\cup, f^T_\cup, f_\Sigma, f^T_\Sigma, f_\#\}$, the relation R_f on $\mathcal{M}_{n\times n}(V^*)^2$ is defined by $(M^{(0)}, M^{(1)}) \in R_f$ if and only if $f(M^{(0)}) = f(M^{(1)})$.

In the experiment for some anonymity notions $\mathbf{N}$ (see Table 1), the attacker should choose two scenarios (as sequences of matrices) and distinguish between them. To prevent the attacker from distinguishing between the scenarios according to information that the anonymity notion does not aim to hide (*unprotected data*), [15] define for the different anonymity notions, relations on the scenario's matrices. Every relation enforces both the matrices to contain the same unprotected data. The relations appear in Table 1.

While all of these notions are applicable to our experiment, in this paper we focus on the strongest relations could be achieved against the different attackers: *unobservability* (UO) and *sender anonymity* (SA).

The unobservability relation $R_{\mathbf{UO}}$ simply holds for all matrices pairs, i.e., does not restrict the matrices at all. The sender anonymity relation $R_{\mathbf{SA}}$ requires that for every (recipient) i, in both the matrices the i-th column contains the

[1] We replaced [15]'s notation $\mathcal{P}(V)$ with V^*, because a powerset does not contains multisets

N	Notion	Definition of $R_{\mathbf{N}}$
SUL	Sender Unlinkabilitiy	$R_{SUL} \overset{def}{=} R_{f_{\Sigma}} \cap R_{f_{\cup}^T}$
RUL	Receiver Unlinkabilitiy	$R_{RUL} \overset{def}{=} R_{f_{\cup}} \cap R_{f_{\Sigma}^T}$
UL	Unlinkabilitiy	$R_{UL} \overset{def}{=} R_{f_{\Sigma}} \cap R_{f_{\Sigma}^T}$
SA	Sender Anonymity	$R_{SA} \overset{def}{=} R_{f_{\cup}^T}$
RA	Receiver Anonymity	$R_{RA} \overset{def}{=} R_{f_{\cup}}$
SA*	Strong Sender Anonymity	$R_{SA*} \overset{def}{=} R_{f_{\Sigma}^T}$
RA*	Strong Receiver Anonymity	$R_{RA*} \overset{def}{=} R_{f_{\Sigma}}$
SRA	Sender-Receiver Anonymity	$R_{SRA} \overset{def}{=} R_{f_{\#}}$
UO	Unobservability	$R_{UO} \overset{def}{=} \mathcal{M}_{n\times n}(V^*)^2$

Table 1: Hevia and Micciancio's [15] table for anonymity variants defines each variant $\mathbf{N}$ and its associated relation $R_{\mathbf{N}}$

same messages. Namely, every participant receives the same messages (the attacker cannot learn information by what the recipients received). That way, the attacker can distinguish between the scenarios only by the senders.

2.2.3 The $R_{\mathbf{N}}^H$ relation

The $R_{\mathbf{N}}$ relations are applicable only for passive adversaries. If the attacker controls a peer in the protocol, it can just inspect the messages in the peer's application queue and check whether they are from $M^{(0)}$ or from $M^{(1)}$. It can do the same also with the messages that the controlled peer receives. Consequently, the $R_{\mathbf{N}}$ relations, cannot be used for active adversaries. We address this by defining new relations family, named $R_{\mathbf{N}}^H \subseteq \mathcal{M}_{n\times n}(V^*)^2$.

Definition 1. *For a given $n \in \mathbb{N}$, consider a pair of matrices, $(M^{(0)}, M^{(1)}) \in \mathcal{M}_{n\times n}(V^*)^2$, $H \subseteq [n]$, and a relation $R_{\mathbf{N}} \subseteq \mathcal{M}_{n\times n}(V^*)^2$. We say that $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}^H$ if and only if*

1. $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}$.
2. *For every $i \in [n] - H$ and $j \in [n]$, $M_{i,j}^{(0)} = M_{i,j}^{(1)} = M_{j,i}^{(0)} = M_{j,i}^{(1)} = \emptyset$.*

The case when messages are sent from honest peers to corrupted, is the case of malicious destination that is discussed in Section 3. $R_{\mathbf{N}}^H$ extends the demand of identical unprotected data in both the matrices, to active attackers. Figure 1 depicts the relation.

2.2.4 Experiment additional notations

$STATE_i$ is the state of the i-th participant. The experiment saves and manages the states of the honest participants.

$STATE_{\mathcal{A}}$ is the state of the attacker $\mathcal{A}$. The experiment gets and saves the attacker state after every action of $\mathcal{A}$, and sends it as a parameter for every action $\mathcal{A}$ should do. The initial information for $\mathcal{A}$ is the initial state of the peers it controls.

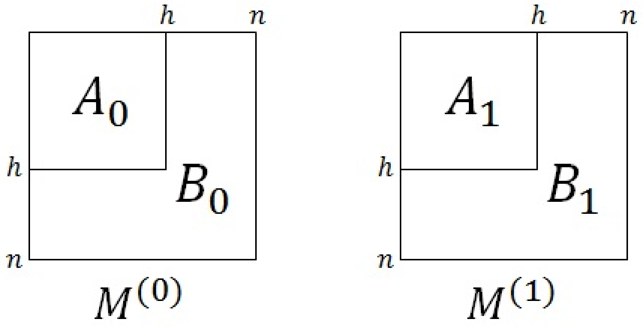

Figure 1: Example of $R_{\mathbf{N}}^H$, for $H = [h] \subset [n]$. $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}^H$ if and only if $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}$ and B_0 and B_1 contain only empty messages multisets. Notice that $(A_0, A_1) \in R_{\mathbf{N}} \subseteq \mathcal{M}_{|H|\times|H|}(V^*)^2$.

We use $C_{i,j,t}$ to denote the set of the elements (possibly ciphertexts), that were sent from the i-th participant to the j-th participants (the participants that are represented by $STATE_i$ and $STATE_j$) during the t-th iteration.

2.3 The Experiment $Expt_{\pi,n,\mathcal{A},Cap}^{\mathbf{Comp-N-b}}(k)$

The experiment (see Algorithm 1) simulates the protocol, π, over one of two scenarios that the attacker, $\mathcal{A}$, chooses and manages adaptively. The simulated scenario is chosen by the b bit. At the beginning of the experiment, π's *setup* produces a sequence of initial states for all the simulation participants. The set of the participants' indexes that $\mathcal{A}$ gets their incoming and outgoing traffic is denote by EV, and the honest peers indexes set is denoted by H. Both the sets are determined by the Cap and n parameters. $\mathcal{A}$ is then initialized with the states of the participants it controls, and decides the maximal number of iterations that the experiment will run ($rounds \in poly(k)$, as $\mathcal{A}$ is a $\mathcal{PPT}$ algorithm, and it writes 1^{rounds}).

During the simulation, the attacker receives all the incoming and outgoing traffic of the controlled and eavesdropped participants. Every experiment's iteration, begins with an option for $\mathcal{A}$, to choose two messages matrices, $M^{(0)}$ and $M^{(1)}$. The experiment verifies that the matrices have identical unprotected data by the tested anonymity notion, $\mathbf{N}$ (verifies that $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}^H$).

If the matrices are valid, the experiment passes only the messages in the $M^{(b)}$ matrix to the application queues of the participants and simulates the honest participants by π. $\mathcal{A}$ simulates the participants it controls (unnecessarily by the protocol).

At the end of every iteration, $\mathcal{A}$ has an opportunity to guess which scenario was simulated. If the attacker does not want to guess, it just returns $NULL$ into b'. When $\mathcal{A}$ chooses the bit b', the experiment is ended with the output b'. The experiment might be ended in returning 0 if the attacker chooses invalid pair of matrices ($\notin R^H_{\mathbf{N}}$), or after *rounds* iterations.

Algorithm 1 $Expt^{\mathbf{Comp-N-b}}_{\pi,n,\mathcal{A},Cap}(k)$

```
for $i = 1$ to $n$ do $S_i \leftarrow \pi.Setup(1^k, i, n)$ end for
$EV = Cap[\overline{H}] \cup Cap[\mathrm{EV}]$
$H = [n] - Cap[\overline{H}]$
$< S_{\mathcal{A}}, 1^{rounds} > \leftarrow \mathcal{A}.Initialize(1^k, \{S_i\}_{i \in Cap[\overline{H}]})$
for $t = 1$ to $rounds$ do
    $< S_{\mathcal{A}}, M^{(0)}, M^{(1)} > \leftarrow \mathcal{A}.InsertMessages(1^k, S_{\mathcal{A}})$
    if $(M^{(0)}, M^{(1)}) \notin R^H_{\mathbf{N}}$ then
        return 0
    end if
    for all $i \in H$ do
        $< S_i, \{C_{i,j,t}\}^n_{j=1} > \leftarrow$
            $\pi.Simulate(1^k, S_i, \{C_{j,i,t-1}\}^n_{j=1}, \{m^{(b)}_{i,j}\}^n_{j=1})$
    end for
    $< S_{\mathcal{A}}, \{C_{i,j,t}\}_{\substack{i \in Cap[\overline{H}] \\ 1 \le j \le n}} > \leftarrow$
        $\mathcal{A}.Simulate(1^k, S_{\mathcal{A}}, \{C_{i,j,t-1}\}_{i \vee j \in EV})$
    $< S_{\mathcal{A}}, b' > \leftarrow A.GuessB(1^k, S_{\mathcal{A}})$
    if $b' \neq NULL$ return $b'$ end if
end for
return 0
```

Definition 2. *The* **Comp-N-advantage** *of an attacker $\mathcal{A}$ that runs with k as a parameter, is defined as:*

$$Adv^{Comp-\mathbf{N}}_{\pi,n,\mathcal{A},Cap}(k) =$$

$$|Pr[Expt^{Comp-\mathbf{N}-1}_{\pi,n,\mathcal{A},Cap}(k) = 1] - Pr[Expt^{Comp-\mathbf{N}-0}_{\pi,n,\mathcal{A},Cap}(k) = 1]|$$

Definition 3. *Protocol π is Comp-$\mathbf{N}$-anonymous, when $\mathbf{N} \in \{SUL, RUL, UL, SA, RA, SA^*, RA^*, SRA, UO\}$, against attackers with capability $Cap \in P([n])^2$, if for all $\mathcal{PPT}$ algorithms, $\mathcal{A}$, there exists a negligible function negl such that,*

$$Adv^{Comp-\mathbf{N}}_{\pi,n,\mathcal{A},Cap}(k) \le negl(k)$$

2.4 Experiment Run Time is $\mathcal{O}(poly(k))$

The total runtime of the experiment (Alg 1) is critical for the proof of the anonymity notions. Using our definition, it is possible to formally prove anonymity notions by a polynomial time reduction to cryptographic primitives. The reduction contains simulation of the above experiment, and therefore its runtime must be polynomial in the security parameter k.

All the actions during the simulation take $\mathcal{O}(poly(k))$, and all the loops run for $\mathcal{O}(poly(k))$ iterations: The algorithms π and $\mathcal{A}$ are polytime, and all the other actions in the experiment take constant time. The main loop in the experiment does no more iterations than the length of a parameter that $\mathcal{A}$ outputs in $poly(k)$ time (during the $Initialize$ method with 1^k as the first argument), such that the loop's iterations can be bounded by some polynomial in k. The inner loop does no more than n iterations, and n is $poly(k)$. The attacker's total runtime is also polynomial in k, as the attacker's total runtime is bounded by the experiment's total runtime.

3. ANONYMITY AGAINST MALICIOUS DESTINATION

The Comp-$\mathbf{N}$-anonymity definition of the previous section covers the following attackers: eavesdroppers, malicious peers, and any combination of them that controls the application adaptively. However, due to the restrictions of the $R^H_{\mathbf{N}}$ relation, the definition cannot be used for testing anonymity properties when the attacker controls destinations of messages from honest peers. Such an attacker model is relevant for anonymous mail services and anonymous web surfing. Namely, this is the main goal of peer to peer networks like Tor [9] and services like Anonymizer [2].

In this section we extend Definition 3 to deal also with malicious destination. This extension is relevant only for two Comp-$\mathbf{N}$-anonymity notions: $\mathbf{N} \in \{SUL, SA\}$ (see Table 1). The other anonymity notions are aimed to hide information that the destination has, and therefore they are irrelevant in such an attacker model.

To extend the definition also against malicious destination, we apply the $R_{\mathbf{N}}$ relation also on the messages from honest peers to malicious. We enforce this new restriction by defining a new relation, $\widehat{R}^H_{\mathbf{N}}$. Figure 2 depicts the new relation.

Definition 4. *($\widehat{R}^H_{\mathbf{N}}$) For a given $n \in \mathbb{N}$, consider a pair of matrices, $(M^{(0)}, M^{(1)}) \in \mathcal{M}_{n \times n}(V^*)^2$, a relation $R_{\mathbf{N}}$ for $\mathbf{N} \in \{SUL, SA\}$, and $H \subseteq [n]$. We say that $(M^{(0)}, M^{(1)}) \in \widehat{R}^H_{\mathbf{N}}$ if and only if*

1. *$(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}$.*
2. *For every $i \in [n] - H$ and $j \in [n]$, $M^{(0)}_{i,j} = M^{(1)}_{i,j} = \emptyset$.*

3.1 Comp-N-Anonymity Against Malicious Destination

We extend the definition to deal with malicious destination, by extending the capability and the Comp-$\mathbf{N}$-b experiment (Alg 1).

3.1.1 Extending the attacker's capability

We add a bit MD to the attacker's capability. This bit indicates whether the attacker is treated as malicious destination or not. After the addition, $Cap = (\overline{H}, EV, MD) \in \mathcal{P}([n])^2 \times \{0, 1\}$. Like the other Cap's elements, we denote Cap's MD by $Cap[MD]$.

[2] www.anonymizer.com.

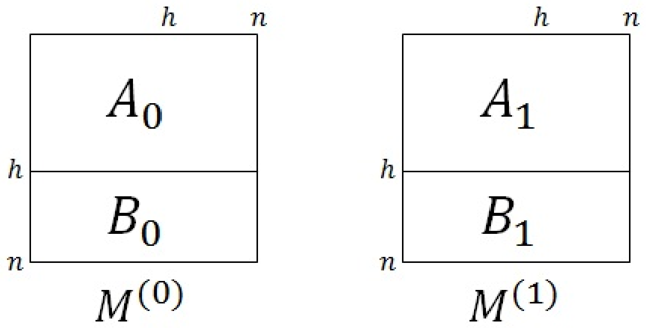

Figure 2: Example of $\widehat{R}^H_{\mathbf{N}}$, for $H = [h] \subset [n]$. $(M^{(0)}, M^{(1)}) \in R^{H,\tau}_{\mathbf{N}}$ if and only if $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}$, and B_0 and B_1 contain only empty messages multisets.

The default value of $Cap[MD]$ is 0, so when testing protocol's anonymity notion not against malicious destination, the capability could be written as before the extension.

3.1.2 Extending the Comp-N-b experiment (Alg 1)

. The messages matrices verification should be done either by $R^H_{\mathbf{N}}$ or by $\widehat{R}^H_{\mathbf{N}}$, according to the attacker capability. The change is in line 7:

if $(Cap[\text{MD}] = 0$ **and** $(M^{(0)}, M^{(1)}) \notin R^H_{\mathbf{N}})$ **or** $(M^{(0)}, M^{(1)}) \notin \widehat{R}^H_{\mathbf{N}}$ **then**.

4. FEASIBILITY OF COMP-N-ANONYMITY AGAINST STRONG ATTACKERS

We say that protocol ensures *ultimate anonymity* if it ensures both the strongest anonymity notions we can require from a protocol:

1. Comp-**SA**-anonymity (sender anonymity) against malicious destination that is a global eavesdropper and controls additional participants (in short: *strong malicious destination*).
2. Comp-**UO**-anonymity (unobservability) against global eavesdropper and malicious peers (*strong attacking peers*).

In order to exclude the trivial solution of the protocol that does not send any message, we limit the discussion to protocols that ensure the *liveness* property; informally, protocols that while the attacker does not deviate from the protocol, ensure messages delivery.

Stronger property we would like to get is *t-liveness*; informally, a protocol satisfies t-liveness if it ensures messages delivery in the presence of up to t malicious participants.

While ensuring unobservability against strong attacking peers is almost trivial, it is complicated to ensure sender anonymity against strong malicious destination and both the demands together (ultimate anonymity).

4.1 Ensuring Comp-UO-Anonymity Against Strong Attacking peers

It is possible to ensure any anonymity notion **N**, against any combination of malicious participants and eavesdroppers (without malicious destination).

We now present a simple protocol that ensures Comp-**UO**-anonymity against strong attacking peers (and therefore ensures all the other anonymity notions; see the technical report [11]). In this protocol, every round, every participant sends a message (real or dummy) to every other participant; the communication is semantically secure encrypted [3].

As in this protocol honest peers communicate with each other directly, no information can be learned about the scenarios without learning information from the encrypted content. Formal proof of this protocol could be done by reducing the Comp-**UO**-anonymity to the security of the encryption scheme.

Protocols that ensure Comp-**UO**-anonymity and yet provides some anonymity (although not Comp-**SA**-anonymity; see below) are DC-net [8] and mixnet [7] based protocols [22] that use constant rate sending.

4.2 Known Protocols are Not Comp-SA-Anonymous Against Strong Malicious Destination

None of the known protocols [21] is a Comp-**SA**-anonymous protocol against strong malicious destination (especially when the attacker controls any minority of the participants). In Theorem 6 we show that such a protocol exists; Theorem 7 shows that any Comp-**SA**-anonymous protocol against malicious destination that is also global eavesdropper, must have high overhead.

As example to hardness of the demand, we bring the DC-net protocol [8] that ensures sender anonymity against the destination by every definition we encountered. We disprove DC-net's Comp-**SA**-anonymity even against passive destination alone in Appendix A.2. Briefly, DC-net fails to hide whether two messages are sent by the same peer or by different two peers.

Similar attack works also against a scheme of many peers that sends via mix or mixes cascade [7]. When the destination controls also some mixes, other attacks are possible [23].

4.3 Ensuring Ultimate Anonymity Against Strong Attackers

It was shown that it is feasible for n parties to compute a polynomially-computable functionality f, without any of them learning anything but the result of the computation, even if some minority of them are malicious. This is possible using techniques of secure multi-party computation [13]. There has been many results in this area, where the most basic ones are of BGW [4, Theorem 3] with malicious minority of less than $\frac{n}{3}$, and of GMW [14] with any malicious minority. In Theorem 6 we prove that although none of the known protocols [21] satisfies ultimate anonymity, there exists a protocol that satisfies both ultimate anonymity and t-liveness for every $t < \frac{n}{2}$. The proof relies on the GMW's theorem [14] (informally in Theorem 5) although for the purpose of feasibility proof, other protocols would be useful as well.

THEOREM 5. (Informal): *Consider a synchronous network with pairwise semantically secure encrypted channels. Then:*

For every polynomially-computable n-ary functionality f, there exists a polynomial time protocol for computing f with computational security in the presence of a malicious adversary corrupting up to $\frac{n}{2}$ of the parties. Namely, every party learns no more than its own inputs and outputs.

THEOREM 6. *There exists protocol Π such that that:*

1. *For every $t < \frac{n}{2}$, Π satisfies t-liveness.*
2. *Π ensures ultimate anonymity; i.e., for every $S \subset [n]$, $|S| < t$, and every $\mathcal{PPT}$ attacker $\mathcal{A}$, $Adv^{Comp-\mathbf{SA}}_{\Pi,n,\mathcal{A},(S,[n],1)}(k)$ and $Adv^{Comp-\mathbf{UO}}_{\Pi,n,\mathcal{A},(S,[n],0)}(k)$ are negligible.*

PROOF. We presents n-ary functionality that given a trusted party, satisfies both the anonymity demands of ultimate anonymity and t-liveness. Relying Theorem 5, this trusted party can be replaced with the n participants such that $t < \frac{n}{2}$ of them are malicious. The functionality is aimed to send up to some $S \in poly(k)$ messages.

For simplicity, we consider a simple scheme of n participants, such that all the participants send anonymous messages only to one of them (the destination).

The n-functionality f is described in Algorithm 2. As an input to the secure computation, every peer chooses the lexicographically-first message in its application queue (or a dummy message if the application queue is empty), and as output the destination receives the lexicographically-lowest message, and the other peers receives empty output.

f has a state that saves all the real messages lexicographically-sorted; we refer this state as a priority queue by lexicographic order, PQ. Because the state must be of constant size (otherwise, the attacker can learn about the number of real messages that were sent), we choose $|PQ| = S$, and to prevent learning from overflows, we limit the number of the delivered messages by the protocol to S (we could do better, but for the feasibility proof S is enough). f is polynomially-computable in the security parameter k (the inputs length and $|PQ|$ are $\in poly(k)$, and the f is polynomial time algorithm).

Our experiment is synchronous, and pairwise semantically secure encrypted channels can be ensured during the setup stage of the experiment (Alg 1). Therefore, from Theorem 5 it is enough to prove that a protocol with trusted party that gets n inputs satisfies both the theorem requirements, when some minority of the inputs is completely controlled by the attacker, and the other (the ones that represent the honest peers) are restricted by the relevant relations.

Algorithm 2 The n-ary functionality f.
State: A priority queue by lexicographic order PQ, and *Counter* for the incoming real messages. The initial state of PQ is an empty priority queue, and *Counter* starts from 0.
Input: n messages $(m_1, m_2, ..., m_n)$.
Output: We denote the destination as the i-th party; the output is $(o_1, o_2, ..., o_i, ..., o_n)$, such that o_i is the first message in the priority queue PQ (or $\perp$ message if PQ is empty), and for every $j \neq i$, $o_j = \perp$.

```
for all message $m$ in $Sort(m_1, m_2, ..., m_n)$ do
    if $m$ is a real message and $Counter < |PQ|$ then
        $PQ.insert(m)$
        $Counter = Counter + 1$
    end if
end for
if $PQ$ is empty then
    $m = dummy$
else
    $m = PQ.removeHead()$
end if
$Output = (\perp)^n$
$Output[i] = m$
return Output
```

4.3.1 Comp-SA-anonymity against strong malicious destination

We prove that given any $S \subset [n]$, $|S| \leq t < \frac{n}{2}$, and a trusted party that calculates the n-ary functionality f (see Alg 2), and that the communication between the trusted party and the peers is secure, it holds that $Adv^{Comp-\mathbf{SA}}_{\Pi,n,\mathcal{A},(S,[n],1)}(k) \leq negl(k)$.

Namely, given n peers, some minority of them is malicious, a trusted party, and a global eavesdropper malicious destination that controls the malicious peers, no $\mathcal{PPT}$ attacker $\mathcal{A}$ can distinguish between any two scenarios with identical unprotected data with non-negligible probability.

In the proof we assume the destination of the messages is one of the malicious peers; otherwise, the proof is as for the unobservability case.

The only information that the attacker receives is the outputs to the (only) malicious destination from the trusted party. We prove that in every two scenarios with the same unprotected data, i.e., two scenarios that are represented by two sequences of messages matrices $\{M_i^{(0)}\}_{i=1}^s$ and $\{M_i^{(1)}\}_{i=1}^s$, such that for every $1 \leq i \leq s$, $(M_i^{(0)}, M_i^{(1)}) \in \widehat{R}^H_{\mathbf{SA}}$, the information that the attacker gets is identical in both the scenarios.

Every round of the protocol, the malicious destination receives one message. We claim that in both the scenarios, it receives exactly the same messages in the same order. Every honest peer sends the lexicographically-first message from its application level that has not sent yet, to the trusted party. Malicious peers might sends whatever they want. Among all these messages, the trusted party sends to the destination the lexicographically-first message. Therefore every round the message with the lowest lexicographic value (from all the application messages that have not reached the destination until this round) is sent to the destination. This happens regardless the distribution of the application messages among the l senders, because the lexicographically-first message among the messages from the honest peers' application level is always sent to the trusted party.

In both the scenarios, due to $\widehat{R}^{H}_{\mathbf{SA}}$, every round the same messages are inserted into the application queue of the honest peers for the destination (the only possible difference is the distribution of the messages among the honest potential sender). The peer with the lexicographically-first message in each scenario will send it to the trusted party.

Because the attacker receives identical information in both the scenarios, it cannot distinguish between the scenarios.

4.3.2 Comp-UO-anonymity against strong attacking peers

Similarly to the Comp-**SA**-anonymity proof, and under the same notions, we need to prove that $Adv^{Comp-\mathbf{UO}}_{\Pi,n,\mathcal{A},(S,[n],1)}(k)$ is not negligible in k for any $\mathcal{A}$.

We assume the attacker does not control the destination; otherwise it is impossible to ensure unobservability. From Theorem 5, the malicious peers cannot learn more than their own inputs and outputs. But their inputs are chosen regardless of the honest peers inputs, and by f (Alg 2) the malicious peers' output is always $\perp$. Consequently, the attacker does not learn any information about the simulated scenario.

4.3.3 t-liveness for $t < \frac{n}{2}$

According Theorem 5, Π ensures delivery of S messages while only some minority of the participants is malicious. We now prove that any honest peer, can ensure message delivery of his own $\lfloor \frac{S}{n} \rfloor$ messages. Π ensures the delivery of the first S messages. Every round, every peer can add one messages to PQ, therefore in the first $\lfloor \frac{S}{n} \rfloor$ rounds every peer can add $\lfloor \frac{S}{n} \rfloor$ messages.

The attacker can limit the honest peers to this number of messages (the relative share of the peer), and can only affect the delay by sending the lexicographically-lowest messages. □

4.3.4 Remarks on the protocol Π (described in the proof)

1. Theorem 7 shows that the throughput of any protocol that satisfies ultimate anonymity cannot be higher than the minimal sending rate of some sender.

2. The lexicographic order of the messages in the application queues and in PQ is necessary. Let Π' be identical protocol, but such that the messages are chosen uniformly out of the application queues, and the trusted party's priority queue (by lexicographic order) is replaced with a multiset of messages, such that the message to send is chosen uniformly among the messages in the multiset.

   We consider the following two scenario: In the first scenario, only p_1 sends the destination $\{m_1, m_1, m_1, m_2\}$, and in the second scenario p_1 sends the destination $\{m_1, m_1, m_1\}$, and p_2 sends the additional m_2. Malicious destination attacker can distinguish between the scenarios by the distribution of the first message that arrives. In the first scenario, the probability of m_2 to reach the destination first is $\frac{1}{4}$, while in the second scenario, the probability of the same event is $\frac{1}{2}$. Consequently, Π' is not Comp-**SA**-anonymous against malicious destination.

4.4 Malicious Destination and Inefficiency

We now prove that the cost of ensuring Comp-**SA**-anonymity against strong malicious destination must be low efficiency (high communication overhead).

We define the number of messages that a peer p_i sends in the first R rounds of a run (scenario) σ of some deterministic protocol by $L_i(\sigma, R)$.

THEOREM 7. *For every deterministic protocol, π, if π is Comp-**SA**-anonymous against malicious destination that is also global eavesdropper, then for every run (scenario) of π, σ and $R \in \mathbb{N}$, during the first R rounds of σ:*

1. *The maximal number of messages that reach the destination is $MaxOut_{\sigma,R} = min\{L_i(\sigma, R)|p_i$ is a honest potential sender$\}$.*

2. *The minimal number of messages that were sent during the first R rounds of σ is $ComOver_{\sigma,R} \geq MaxOut_{\sigma,R} \cdot |\{p_i$ is a honest potential sender$\}|$.*

PROOF. (sketch) Let π be some deterministic protocol that ensures Comp-**SA**-anonymity against malicious destination that is also global eavesdropper. If some participant p_i of π, sends traffic according to the number of messages in its application queue, a global eavesdropper attacker can detect that, by choosing two different scenarios where p_i sends different amount of messages in its application queue.

Therefore, π's participants send regardless the messages in their application queue, and for each participant p_i, for every scenario σ and every R, $L_i(\sigma, R)$ is some constant.

Assume on the contrary that there are some σ' and R' such that $MaxOut_{\sigma',R'} \geq min\{L_i(\sigma', R')|p_i$ is a honest potential sender$\}$. Let $L_i(\sigma', R')$ get minimal value when $i = j$; namely, p_j is the participant with the lowest $L_i(\sigma', R')$ value.

Let σ' be described by $\{M_i^{(0)}\}_{i=1}^{R'}$.

For every $M_i^{(0)}$ matrix we define $M_i^{(1)}$ as follows: for every $i \in [R']$, $l \in [n]$, $M_{i_{j,l}}^{(1)} = \cup_{k=1}^{n} M_{i_{k,l}}^{(0)}$, i.e., p_j sends all the messages that were sent by $M_i^{(0)}$ to the same destinations. Obviously, for every $i \in [R']$, $(M_i^{(0)}, M_i^{(1)}) \in \widehat{R}^{H}_{\mathbf{SA}}$.

We now consider the following run of the Comp-**SA**-b experiment (Alg 1): The attacker simulates the experiment to R' rounds such that every round it chooses $(M_i^{(0)}, M_i^{(1)})$. During the simulation, it acts as a honest participant, but count the messages that reach the malicious destination in some counter C. In the end of the R' rounds, if $C > L_j(\sigma', R')$ the attacker returns 0, and otherwise returns 1.

Because π is deterministic, if $b = 0$ then from the choice of σ' and R', $C >MaxOut_{\sigma',R'} = L_j(\sigma', R')$, and if $b = 1$ then $C \leq L_j(\sigma', R')$, as p_j sent all the messages. Therefore the attacker has the maximal advantage (Definition 2), 1, and π is not Comp-**SA**-anonymous against malicious destination that is also global eavesdropper. In contradiction to the initial assumption.

This proves the first claim of the theorem. The second claim, derived directly from the definition of $L_i(\sigma, R)$ and from the first claim. □

Similar theorem for probabilistic protocols will appear in the full paper. An important observation that follows the above theorem, is that when the peers send independently of each other (must happen in the case of malicious peers), because the maximal output is bounded, the number of the messages in the protocol level increases (and therefore also the used storage).

The above theorem is for protocols that partially satisfy the first demand of ultimate anonymity. Against attackers that satisfy the first demand, and for protocols that satisfy ultimate anonymity, the values of efficiency metrics like maximal output ($MaxOut$), communication overhead ($ComOver$) and latency, are worse. We will formally state and prove the above observations in the full paper.

5. INDISTINGUISHABILITY BETWEEN PERMUTED SCENARIOS

The Comp-**SA**-anonymity definition against malicious destination (see Section 3) is very hard and expensive to achieve, and therefore also ultimate anonymity. The power of malicious destination attacker might seem extremely strong: the attacker chooses the messages to send, affects their timing, and in addition is able to receive these messages and learn information from their arrival times.

This motivates us to create relaxed definition to anonymity notions against malicious destination. Like the extension to the definition in Section 3, this extension is relevant only for two anonymity notions: $\mathbf{N} \in \{SUL, SA\}$.

5.1 Permuted Scenarios

We now present a relaxed relation between the matrices of the messages sent in the two scenarios. We add a restriction on the two chosen scenarios: the only difference between them should be the identities of the senders. We enforce this new restriction, by verifying that for every pair of messages matrices (chosen by the attacker), the rows of the first matrix are some constant permutation of the other.

The same permutation must be used during the whole experiment (we give the attacker to choose it), otherwise some of the problems of the extension in Section 3 arise again. We enforce this new restriction by defining a new relation $R_{\mathbf{N}}^{H,\tau}$ (Definition 8).

Matrix's rows notation. For a matrix $M \in \mathcal{M}_{n\times m}(V^*)$, and for $H \subseteq [n]$, $Rows(M)[H]$ is the set of M's rows with indexes $\in H$.

Definition 8. *For a given $n \in \mathbb{N}$, consider a pair of matrices, $(M^{(0)}, M^{(1)}) \in \mathcal{M}_{n\times n}(V^*)^2$, a relation $R_{\mathbf{N}}$ for $\mathbf{N} \in \{SUL, SA\}$, $H \subseteq [n]$ and a permutation τ over $|H|$ elements. We say that $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}^{H,\tau}$ if and only if*

1. $(M^{(0)}, M^{(1)}) \in \widehat{R}_{\mathbf{N}}^{H}$.
2. $Rows(M^{(0)})[H] = \tau(Rows(M^{(1)})[H])$.

5.2 Comp-$\widehat{\mathrm{N}}$-Anonymity Against Malicious Destination

We denoted the relaxed anonymity notions by $\widehat{\mathbf{N}}$. *Relaxed ultimate anonymity* is ultimate anonymity (see Section 4), but with Comp-$\widehat{\mathbf{SA}}$-anonymity instead of Comp-**SA**-Anonymity. We extend the definition to deal with malicious destination, almost as described in Section 3.1, i.e., the capability is extended, and in the experiment (Alg 1), if $Cap[MD]$=1, the matrix verification in line 7 is done by $R_{\mathbf{N}}^{H,\tau}$ instead of $\widehat{R}_{\mathbf{N}}^{H}$. Additionally, as $\mathcal{A}$ should choose τ, so we add τ to the output arguments of the *Initialize* method (line 4).

5.3 Feasibility of the Permuted Comp-$\widehat{\mathrm{SA}}$-anonymity

Under the new extension, the DC-net protocol [8] in a ring topology, ensures also Comp-$\widehat{\mathbf{SA}}$-anonymity even against malicious destination that is also a global eavesdropper that controls another malicious destination (see Appendix A.3). In more complex topologies, DC-net ensures anonymity even against higher number of malicious peers [27] [12]. In spite of that, DC-net does not ensure t-liveness.

In [11], we present a protocol with communication overhead $O(t^3)$ that ensures relaxed ultimate anonymity when the attacker controls $t < \sqrt{n}$ participants, and also satisfies t-liveness.

6. CONCLUSIONS AND DIRECTIONS

We presented modular definitions covering multiple anonymity notions, against a variety of attackers: eavesdroppers, malicious peers, malicious destination and combinations of them. None of the known protocols [21] satisfies *ultimate anonymity*, i.e., sender anonymity against strong malicious destination *and* unobservability against strong attacking peers; this motivates our study of the feasibility of ultimate anonymity. We proved that there exist a protocol that satisfies ultimate anonymity and also ensures messages delivery, when the attacker controls a minority of the participants. Because *ultimate anonymity* implies inefficiency, we offered relaxed definition to anonymity notions against the destination, that some known protocols like DC-net [8] satisfy.

The first challenge that comes following our work, is to explore the space between protocols that fail to satisfy the ultimate anonymity, and the extremely inefficient protocol (although polynomial) that satisfies it. Namely, to find more efficient protocols that satisfy ultimate anonymity, and better bounds for the efficiency metrics of them. The second challenge is to find the most efficient protocols that ensure relaxed ultimate anonymity, esp., together with robustness requirements.

Another interesting direction is to find bounds for the communication overhead of protocols that satisfy anonymity notions with regarding to the t-liveness property they satisfy. Finally, it would be interesting to explore the implications of relaxing the model, e.g., removing the synchronization assumptions.

7. REFERENCES

[1] M. Backes, I. Goldberg, A. Kate, and E. Mohammadi. Provably secure and practical onion routing. In *Computer Security Foundations Symposium (CSF), 2012 IEEE 25th*, pages 369–385. IEEE, 2012.

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in CryptologyâĂŤCRYPTO'98*, pages 26–45. Springer, 1998.

[3] M. Bellare and P. Rogaway. Asymmetric Encryption. `http://cseweb.ucsd.edu/~mihir/cse207/w-asym.pdf`.

[4] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 1–10. ACM, 1988.

[5] J. Bohli and A. Pashalidis. Relations among privacy notions. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):4, 2011.

[6] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 136–145. IEEE, 2001.

[7] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[8] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of cryptology*, 1(1), 1988.

[9] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium*, pages 303–320. USENIX, 2004.

[10] J. Feigenbaum, A. Johnson, and P. Syverson. Anonymity analysis of onion routing in the universally composable framework.

[11] N. Gelernter and A. Herzberg. On the Limits of Provable Anonymity. Technical Report TR-13-02, Bar Ilan University, April 2013. `http://u.cs.biu.ac.il/~herzbea/security/13-02-ProvAnonLim.pdf`.

[12] S. Goel, M. Robson, M. Polte, and E. Sirer. Herbivore: A scalable and efficient protocol for anonymous communication. 2003.

[13] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*, volume 2. Cambridge university press, 2009.

[14] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, pages 218–229, New York, NY, USA, 1987. ACM.

[15] A. Hevia and D. Micciancio. An indistinguishability-based characterization of anonymous channels. In *Privacy Enhancing Technologies*, pages 24–43. Springer, 2008.

[16] A. Pashalidis. Measuring the effectiveness and the fairness of relation hiding systems. In *Asia-Pacific Services Computing Conference, 2008. APSCC'08. IEEE*, pages 1387–1394. IEEE, 2008.

[17] A. Pfitzmann and M. Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology. *Website, February*, 2008.

[18] A. Pfitzmann and M. Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. *URL: http://dud. inf. tu-dresden. de/literatur/Anon_Terminology_v0*, 34, 2010.

[19] A. Pfitzmann and M. Köhntopp (Hansen). Anonymity, unobservability, and pseudonymityâĂŤa proposal for terminology. In *Designing privacy enhancing technologies*, pages 1–9. Springer, 2001.

[20] M. Reed, P. Syverson, and D. Goldschlag. Anonymous connections and onion routing. *Selected Areas in Communications, IEEE Journal on*, 16(4):482–494, 1998.

[21] J. Ren and J. Wu. Survey on anonymous communications in computer networks. *Computer Communications*, 33(4):420–431, 2010.

[22] K. Sampigethaya and R. Poovendran. A Survey on Mix Networks and Their Secure Applications. *Proceedings of the IEEE*, 94(12):2142–2181, 2006.

[23] A. Serjantov, R. Dingledine, and P. Syverson. From a trickle to a flood: Active attacks on several mix types. In *Information Hiding*, pages 36–52. Springer, 2003.

[24] P. Syverson. Why iâĂŹm not an entropist. In *Security Protocols XVII*, pages 213–230. Springer, 2013.

[25] I. Vajda. Uc framework for anonymous communication. Technical report, Cryptology ePrint Archive Report 2011, 2011.

[26] M. Veeningen, B. De Weger, and N. Zannone. Modeling identity-related properties and their privacy strength. *Formal Aspects of Security and Trust*, pages 126–140, 2011.

[27] M. K. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Transactions on Information and System Security (TISSEC)*, 7(4):489–522, 2004.

APPENDIX

A. DC-NET'S COMP-SA-ANONYMITY AGAINST MALICIOUS DESTINATION

A.1 DC-Net

The dining-cryptographers network protocol [8], is a multi party computation protocol. The protocol is based on Chaum's solution to the dining cryptographers problem: Three cryptographers gather around a table for dinner. The waiter informs them that the meal has been paid by someone, who could be one of the cryptographers or the National Security Agency (NSA). The cryptographers respect each other's right to make an anonymous payment, but want to find out whether the NSA paid. In the solution, every cryptographer flips a coin (or a bit) and shows his result (1 or 0) only to the cryptographer on his left. Now every cryptographer should publish the XOR of his own bit with the bit of the cryptographer on his right side. The cryptographer who paid for the meal (if any) should XOR his result with 1. Now, simply, if the XOR between all the published bits is 0 then NSA paid for the meal, otherwise, it is one of the cryptographers. To send messages of length l, a random bits vector of length l should be chosen. The protocol can be extended to n peers in different topologies, the most common is the ring.

A.2 DC-Net is Not Comp-SA-Anonymous Against Malicious Destination

There is something that the DC-net protocol cannot hide: whether in a round two participant sent or only one. In the DC-net, it takes one round to send a message, and only one participant can send a message in a round (otherwise, there is a collision).

We now consider the following scheme: the three cryptographers p_1, p_2, p_3 want to send anonymous messages to a fourth cryptographer p_4 ($n = 4$). For that purpose, they run the DC-net protocol in rounds between them, and every one of them sends his output to the destination. The destination XORs the three cryptographers output and gets the message.

We present a malicious destination attacker that has non-negligible advantage. The attacker works as follows:

1. In the first round, choose two matrices: in the first scenario p_1 and p_2 send m_1 and m_2 (such that $m_1 \oplus m_2 \notin \{m_1, m_2\}$) respectively, and in the second scenario p_1 sends both the messages (these matrices are legal by $\widehat{R}^{H}_{\mathbf{SA}}$).
2. After the three cryptographers send their first outputs c_1, c_2, c_3, calculate $m' = c_1 \oplus c_2 \oplus c_3$. If $m' \in \{m_1, m_2\}$ return 1. Otherwise return 0.

$\mathcal{A}$ is a polynomial time. And additionally:

$$Adv^{Comp-\mathbf{SA}}_{DC-net,4,\mathcal{A},(\{4\},\emptyset,1)}(k) =$$

$$|Pr[Expt^{Comp-\mathbf{SA}-1}_{DC-net,4,\mathcal{A},(\{4\},\emptyset,1)}(k) = 1]-$$

$$Pr[Expt^{Comp-\mathbf{SA}-0}_{DC-net,4,\mathcal{A},(\{4\},\emptyset,1)}(k) = 1]| = 1$$

Therefore according to the definition of Section 3 DC-net is not Comp-**SA**-anonymous. We note that while the destination does not eavesdrop and does not control some of the peers, collision detection mechanism might be useful. However, such mechanisms might hurt the unobservability of the protocol against malicious peers.

A.3 DC-Net is Comp-$\widehat{\text{SA}}$-Anonymous Against Malicious Destination

We now discuss a scheme of $n > 4$ participants ($n - 1$ potential senders and destination p_n). We give a proof sketch that in a ring topology, while the channels are pairwise encrypted with secure encryption scheme [3], DC-net is Comp-$\widehat{\mathbf{SA}}$-Anonymous by the malicious destination extension of Section 5, even against malicious destination and global eavesdropper attacker that controls additional peer. Namely, an attacker with capability Cap=$(\{i, n\}, [n], 1)$ for some $i \in [n - 1]$.

We omit here the proof for the following claim: given the messages that were sent in a round, and given the final output of all the participants, it is impossible to learn something about the senders identity (unconditional anonymity). This claim holds even if one participant is malicious: i.e., tell the malicious destination what he sent and received [12, Appendix A]. In a ring topology (of more than three peers), for breaking some peer's anonymity, there is a need in both the peers on its sides [27].

Following the above claim, it is enough to prove that if the attacker cannot break the encryption scheme, for every two scenarios with the same unprotected data, in every round in both the scenarios the same messages are sent (scenarios with the same unprotected data are defined by two matrices sequences $\{M_i^{(0)}\}_{i=1}^{s}$ and $\{M_i^{(1)}\}_{i=1}^{s}$, such that for every $1 \leq i \leq s$, $(M_i^{(0)}, M_i^{(1)}) \in R^{H,\tau}_{\mathbf{SA}}$ for some permutation τ over $|H|$ elements).

But by the $R^{H,\tau}_{\mathbf{SA}}$ relation, in every round, when p_j sends m in the first scenario, then $p_{\tau(j)}$ sends m in the second scenario, and therefore the messages that are sent are identical for every round in both the scenarios.

Hence, the attacker cannot break the anonymity without breaking the encryption scheme for learning additional information. Formal proof could be done by reducing the Comp-$\widehat{\mathbf{SA}}$-anonymity to the security of the encryption scheme.