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*. We begin with rigorous definition of anonymity against wide range of computationally bounded attackers, including eavesdroppers, malicious peers, malicious destinations, and their combinations. Following [21], our definition is generic, and captures different notions of anonymity (e.g., unobservability and sender anonymity).

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

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

Keywords

anonymous communication; anonymity; unobservability; unlinkability; metrics; theory

1. INTRODUCTION

Anonymous communication is an important goal, and is also interesting and challenging. Since the publication of the first, seminal paper by Chaum [10], 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 discusses in depth the limitations of this definitional approach [32], 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 [32], we focus on well-defined adversary capabilities, and present a rigorous, indistinguishability-based definition, considering also the strongest possible adversaries and the strongest anonymity requirements.

It seems that 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. Our work extends the definitions of Hevia and Micciancio [21], 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 [21], 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 and/or malicious peers. We extend [21] 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 [21]; 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 [24–26], e.g., unobservability, are trivially inapplicable against a malicious destination (which can observe received traffic). We conclude, that '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 [3]), and between passive and active attackers (for example security against semi honest or malicious adversaries in multi party computation protocols [17]). To deal with adaptive and active attackers, we defined a simulation model for the tested protocols. This challenge was not relevant or addressed in previous works, e.g., [21].

Using our definitions and model, it is possible to formally prove different anonymity notions with respect to different attacker capabilities. The attacker capabilities include the corrupted participants, and the participants to whom it can eavesdrop. Protocols can have different anonymity notions against different attackers with different capabilities. An example to formal proof of anonymity notions against these attackers using our definitions, appears in [15].

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 [21] were limited to eavesdropping attackers, not even ensuring anonymity against the destination; therefore, this is significant, critical extension.

We explore two variants of this definition. The stronger requirements essentially formalizes the strongest anonymity considered in the literature, e.g., in the common terminology [24–26]. 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 [11] that fails to satisfy the stronger requirement, does satisfy this slightly weaker requirement. In Appendix E, we also present improved protocol, which ensures this anonymity property even against multiple malicious nodes.

Organization

In Section 2, we formally define the adversary model, and present our indistinguishability based definition. In Section 3, we extend the definition to consider also malicious destination. In Section 5, we discuss the feasibility of the definition from Section 3 against strong attackers. In Section 6 we present slightly relaxed definition for some of the anonymity notions against malicious destination, and in Section 7 we conclude and discuss future directions.

1.2 Related Work

There is a huge body of research in theory and practice of anonymous communication, beginning with Chaum's paper [10]; see, e.g., a survey of known protocols in [28]. We focus on recent definitional works.

Hevia and Micciancio [21] presented rigorous, indistinguishability based definitions to most anonymity notions, limited eavesdropper adversaries. Our work extends [21] to deal with strong, active, malicious attackers, including destination. Their work contains also detailed discussion of related work until 2008.

Several works use logic based approach, which simplifies the cryptographic properties [19, 20, 22, 33, 35].

Few recent works extend [21] in different ways, e.g., applying the UC framework [8] for anonymous communication [34], and further studying relations among the notions [6,23]. However, these works do not address our goals of studying the strongest anonymity notions, in particular, against strong active adversaries.

Pfitzmann and Hansen offered terminology to anonymity properties [25], that contains comparison between the terminology to the anonymity notions in [21].

Other works offer formal analysis of specific protocols. In [7], [1] and [14] the Onion-Routing (OR) [27] protocol is discussed; the authors present definitions for OR in the UC framework [8]. In [14], the model of Feigenbaum *et al.* [13] is used. In [7] and [1] the authors further discuss the security properties required for OR cryptographic primitives, to achieve provable anonymity.

2. DEFINITIONS

Following Hevia and Micciancio [21], our definition is based on an experiment that simulates protocol runs. We let the adversary choose two scenarios $\{(0), (1)\}$. 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 $\{(0), (1)\}$.

The "relation" between the scenarios is restricted by the anonymity notion $\mathbf{N}$ that is tested in the experiment and by the capabilities of the adversary.

We define two experiments; the first simulates the protocols by the $M^{(0)}$ matrices, and the second by $M^{(1)}$ matrices. The information that the adversary receives during the simulation, is restricted by its capability (for example: global eavesdropper receives all the traffic). The goal of the adversary is to distinguish between the two experiments.

2.1 Network Model, Adversary and Peers

Network model.

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 [10, 29] or routers [12] 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 experiment simulates one of two scenarios that the adversary chooses and manages adaptively. The adversary controls the application layer of all the peers in each of the scenarios, by choosing at every round, two matrices $M^{(0)}, M^{(1)}$ of messages (from each peer, and to each peer). The capabilities of the adversary (e.g., the number of corrupted machines) and the tested anonymity notion, restrict the relation between the scenarios the adversary can choose.

Regardless of these restrictions, peers controlled by the attacker, can deviate arbitrarily from the protocol during the experiment (i.e., act in malicious/byzantine manner).

The ability to select the entire sequence of messages to be sent (the matrices) follows the 'conservative' approach applied in experiments of cryptographic primitives such as encryption [4] [3]. As mentioned in [21], in reality, the attacker might have some influence on the application level of its victims. Because we cannot predict the attacker's influence about the application in different real scenarios, we conservatively give the attacker the whole control.

2.2 Notations and Anonymity Notions

For every type of adversary, and every anonymity notion $\mathbf{N}$ (e.g., anonymity, unlinkability, unobservability) there is a relation that restricts the scenarios that the adversary can choose in the experiment.

Informally, if every efficient adversary of some type (e.g., eavesdropper, or malicious destination), cannot distinguish between every two scenarios, that are restricted by a relation of some anonymity notion $\mathbf{N}$, than the protocol ensures the anonymity notion $\mathbf{N}$ corresponds to the relation, against this kind of adversaries.

In this paper there are four different types of relations:

1. Basic relations $R_{\mathbf{N}}$, introduced in [21], for an adversary that does not control participants (eavesdropper). See Section 2.2.1 and Appendix A.

2. The $R^H_{\mathbf{N}}$ relations extend $R_{\mathbf{N}}$ to deal also with an adversary that controls corrupted participants. See Section 2.2.2.

3. The $\widehat{R}^H_{\mathbf{N}}$ relations extend $R^H_{\mathbf{N}}$, to deal also with malicious destination. See Section 3.

4. The $R^{H,\tau}_{\mathbf{N}}$ relations are a relaxation of $\widehat{R}^H_{\mathbf{N}}$. See Section 6.

Notations.

We use the following common cryptographic and mathematical notations:

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 a set S. Consider a set S, and a multiset $\widehat{S}$, then $\widehat{S} \in S*$ *if and only if* for every $s \in \widehat{S}$, $s \in S$.

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

In this paper, a $\mathcal{PPT}$ algorithm is polynomial time with regarding to its first parameter.

2.2.1 *Eavesdropper adversaries anonymity notions: The $R_{\mathbf{N}}$ relation*

To break anonymity notion $\mathbf{N}$ (see Table 1), the attacker should distinguish between two scenarios it chooses (as sequences of matrices). To prevent eavesdropper attackers from distinguishing between the scenarios according to information that the anonymity notion does not aim to hide (*unprotected data*), [21] define for the different anonymity notions, relations on the scenario matrices. Every relation enforces both the matrices to contain the same unprotected data.

The relations $R_{\mathbf{N}}$ on pairs of scenario matrices for the different anonymity notions, are defined and detailed in Appendix A and in Table 1.

Note that in most of 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 relation for the sender anonymity notion, $R_{\mathbf{SA}}$, requires that for every (recipient) i, in both the matrices the i-th column contains the same messages. Namely, every participant receives the same messages (the attacker cannot learn information by what the recipients receive). That way, the attacker can distinguish between the scenarios only by the senders.

2.2.2 *Anonymity with corrupted nodes: The $R^H_{\mathbf{N}}$ relation*

The $R_{\mathbf{N}}$ relations (Section 2.2.1 and Appendix A) are applicable only for eavesdropper adversaries. If the attacker controls a peer in the protocol, then it can 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 $R^H_{\mathbf{N}}$, such that $H \subseteq [n]$ is the subset of honest participants. The $R^H_{\mathbf{N}}$ relation requires that in addition to the $R_{\mathbf{N}}$ relation on the matrices pair, no messages will be sent (in the application level) from and to malicious participants ($[n] - H$).

$R^H_{\mathbf{N}}$ extends the requirement of identical unprotected data in both the matrices, to active attackers. Figure 1 depicts the $R^H_{\mathbf{N}}$ relation.

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 $(M^{(0)}, M^{(1)}) \in R^H_{\mathbf{N}}$ if and only if*

1. $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}$.

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

Notice that messages are sent from honest peers to corrupted, only in the case of malicious destination; see Section 3.

[1]We replaced [21]'s notation $\mathcal{P}(V)$ with $V*$, because a powerset does not contain multisets.

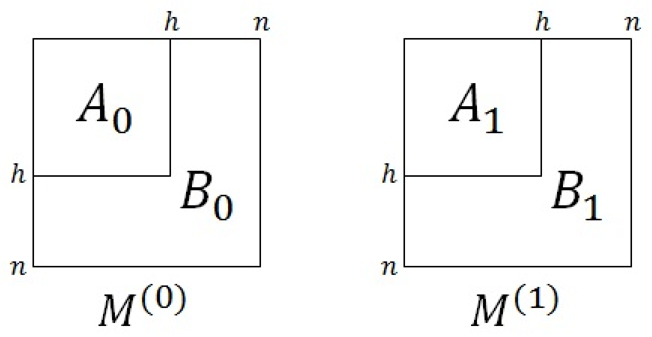

Figure 1: Example of $R^H_{\mathbf{N}}$, for $H = [h] \subset [n]$. $(M^{(0)}, M^{(1)}) \in R^H_{\mathbf{N}}$ *if and only if* $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}$ and B_0 and B_1 contain only empty messages multisets. Notice that for all $R_{\mathbf{N}}$ in Table 1, $(A_0, A_1) \in R_{\mathbf{N}}$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

2.2.3 The attacker capabilities

The attacker capabilities, denoted *Cap*, are a pair $Cap = (Cap[\overline{H}], Cap[\text{EV}]) \in \mathcal{P}([n])^2$. $Cap[\overline{H}]$ specifies the machines controlled by adversary $\mathcal{A}$, and $Cap[EV]$ identifies machines to which the attacker can eavesdrop (e.g., all machines, for a global eavesdropper). An attacker with capability *Cap*, controls the machines with indexes in $Cap[\overline{H}]$ and eavesdrops the traffic of the machines with indexes in $Cap[EV]$.

In Section 3, we extend the attacker capabilities to deal also with malicious destination, by adding to *Cap* another element, a bit $Cap[MD]$; the definition of this section is the same as that definition, using $Cap[\text{MD}] = 0$.

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

2.3.1 The experiment parameters: $\mathbf{N}, b, k, \pi, n, \mathcal{A}$ and *Cap*.

The $\mathbf{N}$ parameter defines the anonymity notion (see Table 1 in Appendix A). $Expt^{\mathbf{Comp-N}-b}_{\pi,n,\mathcal{A},Cap}(k)$ defines a run for $b \in \{0, 1\}$. We test whether a $\mathcal{PPT}$ adversary is able to distinguish between a run where $b = 0$ and a run where $b = 1$.

The parameter k is called security parameter. 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. To initialize the parties, e.g., establish shared (or public/private) keys, we use $\pi.setup$ method, which receives the number of participants n and the identity i of a specific participant as parameters, and outputs the initial state of i (denoted by S_i); this follows the 'common reference string' model [9]. In practice, this simply means that we assume the parties have appropriate keys (shared or public/private). The $\pi.simulate$ 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, and *Cap* is its capabilities.

2.3.2 Overview

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 (line 1), and H is defined to be the honest participants set (line 2). $\mathcal{A}$ 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}. See line 3). The set of the participants' indexes that $\mathcal{A}$ receives their incoming and outgoing traffic during the simulations is $Cap[\overline{H}] \cup Cap[\text{EV}]$ (line 12).

Every experiment's iteration begins with $\mathcal{A}$ choosing two messages matrices, $M^{(0)}$ and $M^{(1)}$ (line 5). The experiment verifies that the matrices have identical unprotected data by the tested anonymity notion, $\mathbf{N}$ (i.e., verifies that $(M^{(0)}, M^{(1)}) \in R^H_{\mathbf{N}}$. See line 6). 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 (line 10). $\mathcal{A}$ simulates the participants it controls (unnecessarily by the protocol. See line 12).

At the end of every iteration, the adversary $\mathcal{A}$ can choose $b' \neq NULL$ to end the experiment and guess b' for the simulated scenario b, or $b' = NULL$ to continue (lines 13 and 14). The experiment might end, returning 0, if the attacker chooses invalid pair of matrices ($\notin R^H_{\mathbf{N}}$), or after *rounds* iterations.

2.3.3 Experiment additional notations

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

$\text{S}_{\mathcal{A}}$ 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 states of the peers it controls.

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 participant (the participants that are represented by S_i and S_j) during the t-th iteration.

2.3.4 Experiment runtime is $\mathcal{O}(poly(k))$

The runtime of the experiment (Alg. 1) is polynomial; this 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.

The main loop in the experiment (lines 4-15) does at most *rounds* iterations, where *rounds* is 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 number of iterations in the main loop is $poly(k)$. The other loops do at most n iterations.

All the actions during the simulation take also polynomial time: The algorithms π and $\mathcal{A}$ are polytime (in the length of the first parameter 1^k), and all the other actions in the experiment take constant time. The attacker's total runtime is also polynomial in k, as the attacker's total runtime is bounded by the experiment's total runtime.

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

```
1: for $i = 1$ to $n$ do $S_i \leftarrow \pi.Setup(1^k, i, n)$ end for
2: $H = [n] - Cap[\overline{H}]$
3: $< S_{\mathcal{A}}, 1^{rounds} > \leftarrow \mathcal{A}.Initialize(1^k, \{S_i\}_{i \in Cap[\overline{H}]})$
4: for $t = 1$ to $rounds$ do
5:     $< S_{\mathcal{A}}, M^{(0)}, M^{(1)} > \leftarrow \mathcal{A}.InsertMessages(1^k, S_{\mathcal{A}})$
6:     if $(M^{(0)}, M^{(1)}) \notin R^H_{\mathbf{N}}$ then
7:         return 0
8:     end if
9:     for all $i \in H$ do
10:        $< S_i, \{C_{i,j,t}\}^n_{j=1} > \leftarrow$
               $\pi.Simulate(1^k, S_i, \{C_{j,i,t-1}\}^n_{j=1}, \{m^{(b)}_{i,j}\}^n_{j=1})$
11:    end for
12:    $< S_{\mathcal{A}}, \{C_{i,j,t}\}_{i \in Cap[\overline{H}], 1 \le j \le n} > \leftarrow$
           $\mathcal{A}.Simulate(1^k, S_{\mathcal{A}}, \{C_{i,j,t-1}\}_{i \vee j \in Cap[\overline{H}] \cup Cap[\mathrm{EV}]})$
13:    $< S_{\mathcal{A}}, b' > \leftarrow A.Guess(1^k, S_{\mathcal{A}})$
14:    if $b' \neq NULL$ return $b'$ end if
15: end for
16: return 0
```

2.4 Comp-N-Anonymity Definition

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 $\mathbf{N} \in \{SUL, RUL, UL, SA, RA, SA*, RA*, SRA, UO\}$ (see Table 1), 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,*

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

3. ANONYMITY AGAINST MALICIOUS DESTINATION

The Comp-**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 one of the main goals of peer to peer networks like Tor [12].

In this section we extend Definition 3 to deal also with malicious destination. This extension is relevant only for two Comp-**N**-anonymity notions: $\mathbf{N} \in \{SUL, SA\}$ (see Table 1 in Appendix A). 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 peers. We enforce this new restriction by defining a new relation, $\widehat{R}^H_{\mathbf{N}}$. Figure 2 depicts the new relation.

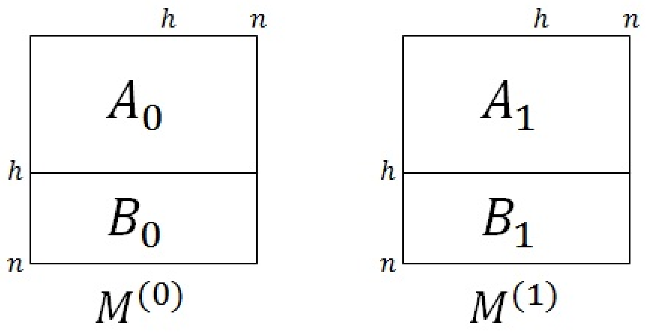

Figure 2: Example of $\widehat{R}^H_{\mathbf{N}}$, for $H = [h] \subset [n]$. $(M^{(0)}, M^{(1)}) \in \widehat{R}^H_{\mathbf{N}}$ *if and only if* $(M^{(0)}, M^{(1)}) \in R_{\mathbf{N}}$, and B_0 and B_1 contain only empty messages multisets.

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-**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, $Cap = (Cap[\overline{H}], Cap[\mathrm{EV}], Cap[\mathrm{MD}]) \in \mathcal{P}([n])^2 \times \{0, 1\}$. Like the other Cap's elements, we denote Cap's MD by $Cap[MD]$.

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 6:

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

4. COMP-N-ANONYMITY DISCUSSION

In this section we discuss the definition of the previous section. We begin with comparison of our definition and the definition of Hevia and Micciancio [21], and then we discuss the Comp-**N**-anonymity of anonymity protocols that rely on high traffic from many users. In the end of this section, we discuss the relations between the different Comp-**N**-anonymity notions, and the anonymity of Comp-**N**-anonymous protocols against attackers with capability Cap, against attackers with other capabilities.

4.1 Our Definition vs. [21]'s Definition

The most striking difference between the definitions, is the kinds of the attackers. While [21] deals only with a passive global eavesdropper, our definition take into consideration wider range of attacker.

Another difference is that in contrast to [21], our definition gives the attacker to control the application level adaptively. In [21]'s experiment, only once, at the beginning of the experiment, the adversary chooses both the matrices. Then the protocol is simulated by one of the matrices, and the adversary should guess which one was simulated. An example that illustrates the difference appears in Appendix C.

4.1.1 [21]'s definition in our model

We now present a short and simple change to our experiment for making [21]'s definition a special case of our definition. The change is additional bit to the attacker's capability, that indicates whether the attacker controls the application adaptively or not. After the addition, $Cap \in \mathcal{P}([n])^2 \times \{0,1\}^2$. If the bit in the capability is 0 (the attacker is non-adaptive), than the experiment gives the attacker to choose the matrices $M^{(0)}$ and $M^{(1)}$ only in the first iteration, and in the rest of the iterations, the experiment fixes the matrices to be empty messages matrices.

In their experiment, Hevia and Micciancio do not specify how to simulate the protocol, but under our protocol's simulation model (see Alg 1) with the new addition, we can test their definition. In their model the attacker is a passive global eavesdropper that does not control the application adaptively; so, we just specify the attacker's capability to be ($\emptyset$, $[n]$, 0, 0). π's setup method of the simulated protocol, returns initial states of n machines.

4.2 Traffic Based Anonymity

The definition of Comp-**N**-Anonymity, gives the attacker the power to control the application level of all the protocol's participants. Therefore, anonymity protocols that their anonymity mainly depends on high traffic from many users, are usually not Comp-**N**-anonymous, even for the weaker **N**-anonymity notions, and against weak attackers.

In Appendix D, we bring a detailed example of a simplified version of the Tor protocol [12]. Tor is the most popular anonymity network ever, and it provides anonymity for millions of users. We show that Tor is not Comp-**N**-Anonymous, for any **N** (see Table 1), even against the weakest attacker: a local eavesdropper to one arbitrary Tor router.

4.3 Relations Between the Comp-N-Anonymity Properties

*4.3.1 Relations between Comp-**N**-anonymity notions*

Similarly to [21], some Comp-**N**-anonymity notions imply other, against attackers with the same capability: UO → SRA → SA* → SA → SUL, SRA → RA* → RA → RUL, SA* → UL → RUL and RA* → UL → SUL. This stems directly from the definition of the $R^H_{\mathbf{N}}$ relation; for every attacker's capability Cap, for every relation above of the form X → Y , $R^H_{\mathbf{Y}} \subset R^H_{\mathbf{X}}$. Hence, an attacker $\mathcal{A}$ that has non-negligible advantage $Adv^{Comp-\mathbf{Y}}_{\pi,n,\mathcal{A},Cap}(k) > negl(k)$, also have non-negligible advantage for the Comp-**X**-anonymity notion.

4.3.2 Relations between the different attacker's capabilities

We first show that as expected, enlarging $Cap[EV]$, the attacker's eavesdropped participants set, can only increase the attacker's power.

LEMMA 5. *If π is a Comp-**N**-anonymous protocol (for some anonymity notion **N** [21]) against attackers with capability Cap = ($\overline{H}$, EV, MD), then π is also Comp-**N**-anonymous against attackers with capability Cap' = ($\overline{H}$, $EV' \subset EV$, MD).*

PROOF. (Sketch) We briefly show how to build an attacker $\mathcal{A}$ with capability Cap that breaks π's Comp-**N**-anonymity, given an attacker $\mathcal{A}'$ with capability Cap' that can do that.

$\mathcal{A}$ runs in one of the Comp-**N**-b experiments ($b \in \{0,1\}$). It then simulates $\mathcal{A}'$ over the experiment, and does exactly what $\mathcal{A}'$ does. $\mathcal{A}$ that gets information according to Cap, pass information to $\mathcal{A}'$, only according to Cap'. The correctness of the reduction is trivial. □

In contrast, enlarging $Cap[\overline{H}]$, the attacker's controlled participants set, might detract its power. The intuition for the counter-example we bring here, is that controlling participants has some price; the $R^H_{\mathbf{N}}$ and $\widehat{R}^H_{\mathbf{N}}$ relations, restrict the controlled participants ($Cap[\overline{H}]$).

LEMMA 6. *If π is a Comp-**N**-anonymous protocol (for some anonymity notion **N** [21]) against attackers with capability Cap = ($\overline{H}$, EV, MD), then π* **not necessarily** *Comp-**N**-anonymous against attackers with capability Cap' = ($\overline{H}' \subset \overline{H}$, EV, MD).*

PROOF. (Sketch) Every protocol is Comp-**N**-Anonymous against attacker that controls all the participants ($\overline{H} = [n]$), because in such a case, there is no protected data at all (from the attacker), and both the messages matrices must be identical. But, there are protocols that are not Comp-**N**-anonymous even against *one* controlled participant. The example in Appendix D, when the attacker controls one router, instead of eavesdropping to it, demonstrates this (for every **N**). □

The last lemma shows that for $\mathbf{N} \in \{SUL, SA\}$, extending the capability of an attacker to be also malicious destination, does not detract the attacker's power.

LEMMA 7. *For $\mathbf{N} \in \{SUL, SA\}$, if π is a Comp-**N**-anonymous protocol against attackers with capability Cap = ($\overline{H}$, EV, 1), then π is also Comp-**N**-anonymous against attackers with capability Cap' = ($\overline{H}$, EV, 0).*

PROOF. Directly from the definitions of the $R^H_{\mathbf{N}}$ and $\widehat{R}^H_{\mathbf{N}}$ relations: for every $\mathbf{N} \in \{SUL, SA\}$ and $H \subseteq [n]$, $R^H_{\mathbf{N}} \subseteq \widehat{R}^H_{\mathbf{N}}$. □

5. ULTIMATE ANONYMITY

DEFINITION 8. *A protocol ensures* ultimate anonymity *if it ensures both the strongest anonymity notions feasible:*

1. *Comp-**SA**-anonymity (sender anonymity) against malicious destination that is a global eavesdropper and controls (a minority of the) additional participants (in short:* strong malicious destination*).*

2. *Comp-*$\mathbf{UO}$*-anonymity (unobservability) against global eavesdropper and (a minority of the) participants (*strong attacking peers*).*

In order to exclude the trivial solution of a 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.

A stronger property we would like to achieve 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 requirements together (ultimate anonymity).

5.1 Ensuring Comp-UO-Anonymity Against Strong Attacking peers

It is possible to ensure any anonymity notion $\mathbf{N}$, against any combination of malicious participants and eavesdroppers (without malicious destination).

In fact, the following trivial protocol ensures Comp-$\mathbf{UO}$-anonymity against strong attacking peers (and therefore ensures all the other anonymity notions; see Section 4.3.1). In this protocol, every round, every participant sends a message (real or dummy) to every other participant; the communication is semantically secure encrypted [4].

Because in this trivial 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 would reduce the Comp-$\mathbf{UO}$-anonymity to the security of the trivial protocol's encryption scheme.

Protocols that ensure Comp-$\mathbf{UO}$-anonymity and yet provide some anonymity (although not Comp-$\mathbf{SA}$-anonymity; see below) are DC-net [11] and mixnet [10] based protocols [29] that use fixed sending rate.

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

None of the known protocols [28] is Comp-$\mathbf{SA}$-anonymous against strong malicious destination.

For example, consider the DC-net protocol [11]. We show that DC-net is not Comp-$\mathbf{SA}$-anonymous even against passive destination (in Appendix B.2). Briefly, DC-net fails to hide whether two messages are sent by the same peer or by different peers.

A similar attack works also against a scheme of many peers that send via mix or mixes cascade [10]. When the destination controls also some mixes, other active attacks are possible [30].

In Theorem 10 we show that a Comp-$\mathbf{SA}$-anonymous protocol exists; on the other hand, Theorems 13 and 15 show that any Comp-$\mathbf{SA}$-anonymous protocol against malicious destination that is also global eavesdropper, must have high overhead.

5.3 Ensuring Ultimate Anonymity Against Strong Attackers

Secure multi-party computation [17] allows n parties to compute a polynomially-computable n-ary functionality f (functionality with n inputs and n outputs), without any of them learning anything but the result of the computation, even if some minority of them are malicious. There has been many results in this area, where the most basic ones are of BGW [5, Theorem 3] with malicious minority of less than $\frac{n}{3}$, and of GMW [18] with any malicious minority. Based on these results, in Theorem 10 we prove that there exists a polynomial time protocol that satisfies ultimate anonymity. The proof relies on the GMW's theorem [18] (informally in Theorem 9) although for the purpose of feasibility proof, other protocols would be useful as well.

THEOREM 9. *(*GMW theorem [18] - 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 Π^f for computing f with computational security in the presence of a malicious adversary corrupting up to $\frac{n}{2}$ of the parties.

Namely, during a run of Π^f, a party might learn more than the case where a trusted third party (TTP) *securely receives f's inputs, computes f, and sends f's outputs to the participants securely, only with negligible probability.*

THEOREM 10. *There exists polynomial time protocol that satisfies liveness and ensures ultimate anonymity (Definition 8).*

PROOF. (sketch) We present a polynomially-computable n-ary functionality f that given a trusted third party (TTP), satisfies ultimate anonymity and liveness. From Theorem 9, this trusted third party can be replaced with a secure multiparty computation protocol Π^f, if the malicious participants set $S \subset [n]$, is a minority of the n participants ($|S| < \frac{n}{2}$). The functionality f sends up to some $c \in poly(k)$ messages.

For simplicity, in this proof sketch, 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 receive empty output.

The functionality 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 $|PQ| = c$, and to prevent learning from overflows, we limit the number of messages delivered by the protocol to c. The functionality f is polynomially-computable in the security parameter k (the inputs length and $|PQ|$ are $\in poly(k)$, and f is a polynomial time algorithm).

From Theorem 9, it is enough to prove that a protocol I^f, with a TTP that receives n inputs, satisfies both the requirements of the theorem, 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.

We next present such protocol I^f. During a run of I^f, the TTP iteratively securely receives f's inputs from the n

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 $j \neq i$, $o_j = \perp$.

```
 1: for all message $m$ in $Sort(m_1, m_2, ..., m_n)$ do
 2:   if $m$ is a real message and $Counter < |PQ|$ then
 3:     $PQ.insert(m)$
 4:     $Counter = Counter + 1$
 5:   end if
 6: end for
 7: if $PQ$ is empty then
 8:   $m = dummy$
 9: else
10:   $m = PQ.removeHead()$
11: end if
12: $Output = (\perp)^n$
13: $Output[i] = m$
14: return Output
```

parties, computes f, and securely sends f's outputs to the parties.

Lemmas 11 (I^f is Comp-**SA**-anonymous) and 12 (I^f is Comp-**UO**-anonymous) together show that I^f ensures ultimate anonymity. The lemmas are stated and proven later in this section.

From Theorem 9, the trusted third party in I^f can be replaced by a polynomial time secure multiparty computation of the n participants, such that the malicious participants set is $S \subset [n]$, and $|S| < \frac{n}{2}$. This protocol is denoted by Π^f, and we assume that during its setup process, every two participants share unique key to construct private channels. Theorem 9 also says that it is impossible for a $\mathcal{PPT}$ $\mathcal{A}$ to learn about the inputs and the outputs of the honest peers from a run of Π^f more than it learns from a run of I^f (except with negligible probability).

Assume on the contrary that $Adv^{Comp-\mathbf{SA}}_{\Pi^f,n,\mathcal{A},(S,[n],1)}(k)$ is not negligible in k.

If $\mathcal{A}$ uses the same inputs for the honest parties during the experiment in both the scenarios, $\mathcal{A}$ cannot achieve any advantage. Hence, $\mathcal{A}$ succeeds to distinguish between two different scenarios with different inputs to the functionality. But from Lemma 11, there is no $\mathcal{PPT}$ $\mathcal{A}$ that can learn such information from a run of I^f, except with negligible probability. Contradiction to Theorem 9, hence Π^f ensures Comp-**SA**-anonymity against strong malicious destination.

Similarly, from Lemma 12, we get that Π^f is also Comp-**UO**-anonymous against strong attacking peers.

To complete the proof, the liveness property of Π^f is derived directly from the correctness of Theorem 9. $\square$

We now prove Lemmas 11 and 12 about the ideal protocol, I^f, described in the proof of Theorem 10.

LEMMA 11. *The I^f protocol ensures Comp-**SA**-anonymity against strong malicious destination.*

PROOF. We prove that given any $S \subset [n]$, $|S| < \frac{n}{2}$, and a trusted third 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 $Adv^{Comp-\mathbf{SA}}_{I^f,n,\mathcal{A},(S,[n],1)}(k) \leq negl(k)$ (notice that we do not count the TTP as a participant).

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

In the proof we assume that the destination of the messages is one of the malicious peers; otherwise, this is the case of strong attacking peers (Lemma 12).

Because I^f uses trusted third party, the only information that the attacker receives is the outputs to the (only) malicious destination from the trusted party. We prove that for 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 receives is identical in both the scenarios.

Every round of the protocol, the malicious destination receives one message (the other malicious peers always receive $\perp$). We claim that in both 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 TTP. Malicious peers might sends whatever they want. Among all these messages, the TTP 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, and the messages of the malicious participants) is sent to the destination.

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 senders). The peer with the lexicographically-first message in each scenario will send it to the TTP, so the TTP always holds the application message with the lexicographically lowest value, that has not been sent yet to the destination. The TTP will send the destination either this message or a message from malicious peer (or a dummy).

Because the adversary receives identical information in both the scenarios, it cannot distinguish between them. $\square$

LEMMA 12. *The I^f protocol ensures Comp-**UO**-anonymity against strong attacking peers.*

PROOF. Similarly to the Comp-**SA**-anonymity proof, and under the same notations, we need to prove that the advantage $Adv^{Comp-\mathbf{UO}}_{I^f,n,\mathcal{A},(S,[n],0)}(k)$ is negligible in k for any $\mathcal{A}$.

All the information the malicious peers receive is just $\perp$ from the TTP (by f's definition, Alg. 2), regardless of the chosen scenarios, and without any option to learn something about the inputs of the honest peers. Consequently, the attacker does not learn any information about the simulated scenario and has no advantage at all. $\square$

5.3.1 *Remark about the protocols Π^f and I^f*

The lexicographic order of the messages in the application queues and in PQ is necessary. Let I' be identical protocol, but such that the messages are chosen uniformly out of the application queues, and the trusted third 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 (valid by the $\widehat{R}^{H}_{\mathbf{SA}}$ relation) scenarios: In the first scenario, only p_1 sends four messages $\{m_1, m_1, m_1, m_2\}$ to the destination, and in the second scenario p_1 sends to the destination $\{m_1, m_1, m_1\}$, and p_2 sends the additional message m_2. A malicious destination 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, I' is not Comp-**SA**-anonymous against malicious destination.

5.4 Malicious Destination and Inefficiency

The protocol we presented in the proof of Theorem 10 satisfies ultimate anonymity, but has very high communication overhead. We now prove that the cost of ensuring Comp-**SA**-anonymity against strong malicious destination must be low efficiency (high communication overhead).

Theorems 13 and 15 discuss the inefficiency of *deterministic* and *probabilistic* protocols (respectively) that ensure Comp-**SA**-anonymity against strong malicious destination.

We define the number of 'send' events by a peer p_i in the first R rounds of a run of a deterministic protocol π, according to scenario σ, by $L^{\pi}_{i}(\sigma, R)$ (this is while the adversary does not deviate from the protocol).

THEOREM 13. *For every deterministic protocol, π, if π is Comp-**SA**-anonymous against global eavesdropper destination, then for every run of protocol π by scenario σ, and every $R \in \mathbb{N}$, during the first R rounds of the run:*

1. *The number of messages that reach the destination is $Out^{\pi}_{\sigma,R} \leq min\{L^{\pi}_{i}(\sigma, R) | p_i \text{ is a honest potential sender}\}$.*
2. *The total number of 'send' events is*
   $Com^{\pi}_{\sigma,R} \geq Out^{\pi}_{\sigma,R} \cdot |\{p_i \text{ is a honest potential sender}\}|$.

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 has different amount of messages in its application queue.

Therefore, for every two scenarios restricted by $\widehat{R}^{H}_{\mathbf{SA}}$, π's participants send regardless the messages in their application queue. Namely, for every p_i, for every two scenarios σ_0, σ_1 (restricted by $\widehat{R}^{H}_{\mathbf{SA}}$) and every R, $L^{\pi}_{i}(\sigma_0, R) = L^{\pi}_{i}(\sigma_1, R)$.

Assume on the contrary that there are some σ_0 and R' such that $Out^{\pi}_{\sigma_0,R'} > min\{L^{\pi}_{i}(\sigma_0, R') | p_i$ is a honest potential sender$\}$. Let $L^{\pi}_{i}(\sigma_0, R')$ get minimal value when $i = j$; namely, p_j is the participant with the lowest $L^{\pi}_{i}(\sigma_0, R')$ value.

Let σ_0 be described by $\{M^{(0)}_{i}\}^{R'}_{i=1}$.

We define a scenario σ_1 by a sequence of matrices $\{M^{(0)}_{i}\}^{R'}_{i=1}$ as follows: for every $i \in [R']$, $l \in [n]$, $M{i}^{(1)}_{j,l} = \cup^{n}_{k=1} M{i}^{(0)}_{k,l}$, i.e., p_j sends all the messages that were sent by $M^{(0)}_{i}$ to the same destinations. Obviously, for every $i \in [R']$, $(M^{(0)}_{i}, M^{(1)}_{i}) \in \widehat{R}^{H}_{\mathbf{SA}}$. Therefore, for every p_i, and in particular for $i = j$: $L^{\pi}_{i}(\sigma_0, R') = L^{\pi}_{i}(\sigma_1, R')$.

We now consider the following run of the Comp-**SA**-b experiment (Alg. 1): The attacker simulates the experiment for R' rounds such that every round i, it chooses $(M^{(0)}_{i}, M^{(1)}_{i})$. During the simulation, the attacker acts as a honest participant, but counts the messages that reach the malicious destination in some counter C. In the end of the R' rounds, if $C > L^{\pi}_{j}(\sigma_0, R')$, then the attacker returns 0, and otherwise returns 1.

Because π is deterministic, if $b = 0$ then from the choice of σ_0 and R', $C = Out^{\pi}_{\sigma_0,R'} > L^{\pi}_{j}(\sigma_0, R')$, and if $b = 1$ then $C \leq L^{\pi}_{j}(\sigma_1, R') = L^{\pi}_{j}(\sigma_0, R')$, as p_j sends all the messages in the scenario σ_1. 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 follows directly from the definition of $L^{\pi}_{i}(\sigma, R)$ and from the first claim. $\square$

To state similar theorem for probabilistic protocols (Theorem 15), we define the *average* number of messages that a peer p_i sends in the first R rounds of in a random run according to scenario σ of some protocol by $\overline{L^{\pi}_{i}}(\sigma, R)$.

To make the proof clearer, we also extend the different relations (see Sections 2.2.1, 2.2.2, 3 and 6) from pair of matrices, to *scenarios*, i.e., pair of matrices sequences.

DEFINITION 14. *For every $n \in \mathbb{N}$, for every relation $R \subseteq \mathcal{M}_{n\times n}(V^{*})^2$, and for every two scenarios, represented by two matrices sequences, $\sigma_0 = \{M^{(0)}_{i}\}^{r_0}_{i=1}$ and $\sigma_1 = \{M^{(1)}_{i}\}^{r_1}_{i=1}$, we say that $(\sigma_0, \sigma_1) \in R$,* if and only if

1. *$r_0 = r_1$.*
2. *For every $i \in [r_0]$, $(M^{(0)}_{i}, M^{(1)}_{i}) \in R$.*

THEOREM 15. *For every probabilistic protocol, π, if π is Comp-**SA**-anonymous against global eavesdropper destination, then for a random run of π according to scenario σ and every $r \in poly(k)$, during the first r rounds of the run:*

1. *The* average *number of messages that reach the destination is $\overline{Out^{\pi}_{\sigma,R}} \leq \min\{\overline{L^{\pi}_{i}}(\sigma, r) | p_i \text{ is a honest potential sender}\} + NEGL(k)$.*
2. *The total number of 'send' events is*
   $\overline{Com^{\pi}_{\sigma,r}} \geq \overline{Out^{\pi}_{\sigma,r}} \cdot |\{p_i \text{ is a honest potential sender}\}| - NEGL(k)$.

PROOF. (sketch) Let π be some probabilistic protocol that ensures Comp-**SA**-anonymity against malicious destination that is also global eavesdropper.

LEMMA 16. *For every two scenarios of π, $(\sigma_0, \sigma_1) \in \widehat{R}^{H}_{\mathbf{SA}}$, every $i \in [n]$, and every $r \in poly(k)$,*
$|\overline{L^{\pi}_{i}}(\sigma_0, r) - \overline{L^{\pi}_{i}}(\sigma_1, r)| \in NEGL(k)$

PROOF. Consider two different scenarios of π, $(\sigma_0, \sigma_1) \in \widehat{R}^{H}_{\mathbf{SA}}$. If the distributions of the number of 'send' events by some participant p_i during the first R iterations (for some $R \in \mathcal{O}(poly(k))$) in both the scenarios, are computationally distinguishable, an attacker can also distinguish between these two scenarios in the Comp-**SA** experiment. Therefore, for every participant of π, the distributions of 'send' events for any two scenarios restricted by the $\widehat{R}^H_{\mathbf{SA}}$ relation, are indistinguishable.

Consequently, for every participant p_i, for every two scenarios $(\sigma_0, \sigma_1) \in \widehat{R}^H_{\mathbf{SA}}$, for every $r \in poly(k)$, it holds that $|\overline{L}^\pi_i(\sigma_0, r) - \overline{L}^\pi_i(\sigma_1, r)| \in NEGL(k)$; otherwise, it is possible to distinguish between the distributions of the 'send' events by inspecting the number of 'send' events of p_i. $\square$

Assume on the contrary that there is some σ_0 and $r' \in poly(k)$ such that $\overline{Out}^\pi_{\sigma_0,r'} > \min\{\overline{L}^\pi_i(\sigma_0, r') | p_i$ is a honest potential sender$\} + NEGL(k)$.

Let $\overline{L}^\pi_i(\sigma_0, r')$ get minimal value when $i = j$; namely, p_j is the participant with the lowest $\overline{L}^\pi_i(\sigma_0, r')$ value.

Let σ_0 be described by $\{M^{(0)}_i\}^s_{i=1}$. $s \geq r'$.

For every $M^{(0)}_i$ matrix we define a matrix for a new scenario σ_1 by $M^{(1)}_i$ as follows: for every $i \in [s]$, $l \in [n]$, $M^{(1)}_{i_{j,l}} = \uplus^n_{k=1} M^{(0)}_{i_{k,l}}$, i.e., p_j sends all the messages that were sent by $M^{(0)}_i$ to the same destinations. Obviously, $(\sigma_0, \sigma_1) \in \widehat{R}^H_{\mathbf{SA}}$ (Definition 14).

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 i it chooses $(M^{(0)}_i, M^{(1)}_i)$. During the simulation, it acts as a honest participant, but count the messages that reach the malicious destination in some counter C.

We argue, that the attacker can distinguish between the experiments (i.e., guess b), by the distribution of the messages that reaches the destination.

Because in scenario σ_1, only p_j sends messages, $\overline{Out}^\pi_{\sigma_1,r'} \leq \overline{L}^\pi_j(\sigma_1, r')$.

Because $(\sigma_0, \sigma_1) \in \widehat{R}^H_{\mathbf{SA}}$, according to Lemma 16, $|\overline{L}^\pi_j(\sigma_1, r') - \overline{L}^\pi_j(\sigma_0, r')| \in NEGL(k)$.

Because $\overline{L}^\pi_j(\sigma_0, r') = \min\{\overline{L}^\pi_i(\sigma_0, r') | p_i$ is a honest potential sender$\}$, from the contradiction assumption we get
$\overline{Out}^\pi_{\sigma',r'} > \overline{L}^\pi_j(\sigma_0, r') + NEGL(k)$.

Therefore $\overline{Out}^\pi_{\sigma_0,r'} > \overline{Out}^\pi_{\sigma_1,r'} + NEGL(k)$, which means that the distributions of the messages that reaches the destination during r' iterations in the scenarios σ_0 and σ_1, are computationally distinguishable (the difference between the averages is not negligible). The attacker can distinguish between the scenarios $\sigma_0, \sigma_1 \in \widehat{R}^H_{\mathbf{SA}}$ in the Comp-**SA**-b experiment with non-negligible probability by these distributions. Contradiction to the fact that π is Comp-**SA**-anonymous.

This proves the first claim of the theorem. The second claim, derived directly from the definition of $\overline{L}^\pi_i(\sigma, r)$ and from the first claim.

An important observation that follows from the above theorems, is that when 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 requirement of ultimate anonymity. Additionally, the adversary in the proof is only passive global eavesdropper destination. For protocols that satisfy ultimate anonymity, the values of efficiency metrics like maximal throughput (Out), communication overhead (Com) and latency, are worse.

6. 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: $\mathbf{N} \in \{SUL, SA\}$.

6.1 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 $R^{H,\tau}_{\mathbf{N}}$ (Definition 17).

Matrix row notation. For a matrix $M \in \mathcal{M}_{n\times m}(V^*)$, we denote the ith row of M by $Row_i(M)$.

DEFINITION 17. *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's elements. We say that $(M^{(0)}, M^{(1)}) \in R^{H,\tau}_{\mathbf{N}}$ if and only if*

1. $(M^{(0)}, M^{(1)}) \in \widehat{R}^H_{\mathbf{N}}$.
2. $Row_i(M^{(0)}) = Row_{\tau(i)}(M^{(1)})$.

6.2 Comp-$\widehat{\text{N}}$-Anonymity Against Malicious Destination

We denoted the relaxed anonymity notions by $\widehat{\mathbf{N}}$. *Relaxed ultimate anonymity* is ultimate anonymity (see Section 5), 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 6 is done by $R^{H,\tau}_{\mathbf{N}}$ instead of $\widehat{R}^H_{\mathbf{N}}$. Additionally, as $\mathcal{A}$ should choose τ, we add τ to the output arguments of the $Initialize$ method (line 3).

6.3 Feasibility of the Permuted Comp-$\widehat{\text{SA}}$-anonymity

Under the new extension, the DC-net protocol [11] in a ring topology, ensures also Comp-$\widehat{\mathbf{SA}}$-anonymity even against global eavesdropper destination that also controls another malicious peer (see Appendix B.3). In more complex topologies, DC-net ensures anonymity even against higher number of malicious peers [36] [16]. In spite of that, DC-net does not ensure t-liveness.

In Appendix E, 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.

7. 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.

Known protocols [28] do not satisfy *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 [11] 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.

8. ACKNOWLEDGMENTS

This research was supported by grants from the Ministry of Science and Technology, Israel, and the Israeli Science Foundation.

9. 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. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *Advances in CryptologyâĂŤASIACRYPT 2001*, pages 566–582. Springer, 2001.

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

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

[5] 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.

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

[7] J. Camenisch and A. Lysyanskaya. A formal treatment of onion routing. In *Advances in Cryptology–CRYPTO 2005*, pages 169–187. Springer, 2005.

[8] 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.

[9] R. Canetti, R. Pass, and A. Shelat. Cryptography from sunspots: How to use an imperfect reference string. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 249–259. IEEE, 2007.

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

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

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

[13] J. Feigenbaum, A. Johnson, and P. Syverson. A model of onion routing with provable anonymity. In *Financial Cryptography and Data Security*, pages 57–71. Springer, 2007.

[14] J. Feigenbaum, A. Johnson, and P. Syverson. Probabilistic analysis of onion routing in a black-box model. *ACM Trans. Inf. Syst. Secur.*, 15(3):14:1–14:28, Nov. 2012.

[15] N. Gelernter and A. Herzberg. Efficient Unobservable Anonymous Reporting against Strong Adversaries. Cryptology ePrint Archive, Report 2013/534, 2013. `http://eprint.iacr.org/`.

[16] S. Goel, M. Robson, M. Polte, and E. G. Sirer. Herbivore: A Scalable and Efficient Protocol for Anonymous Communication. Technical Report 2003-1890, Cornell University, Ithaca, NY, February 2003.

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

[18] 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.

[19] I. Goriac. An epistemic logic based framework for reasoning about information hiding. In *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*, pages 286–293. IEEE, 2011.

[20] J. Halpern and K. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3):483–514, 2005.

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

[22] D. Hughes and V. Shmatikov. Information hiding, anonymity and privacy: a modular approach. *Journal of Computer Security*, 12(1):3–36, 2004.

[23] 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.

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

[25] 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.

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

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

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

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

[30] 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.

[31] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[32] P. Syverson. Why I'm not an entropist. In *Security Protocols XVII*, pages 213–230. Springer, 2013.

[33] Y. Tsukada, K. Mano, H. Sakurada, and Y. Kawabe. Anonymity, privacy, onymity, and identity: A modal logic approach. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 3, pages 42–51. IEEE, 2009.

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

[35] 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.

[36] 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. HEVIA AND MICCIANCIO'S ANONYMITY NOTIONS

Hevia and Micciancio's anonymity notions are defined by the $R_{\mathbf{N}}$ relations [21]. The $R_{\mathbf{N}}$ relations enforce the unprotected data (the attacker assumed knowledge) in a pair of scenario matrices to be identical, with regarding to anonymity notion $\mathbf{N}$. The different relations are 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, they define $f_\cup^T(M) \overset{def}{=} f_\cup(M^T)$ and similarly $f_\Sigma^T(M) \overset{def}{=} f_\Sigma(M^T)$.

For a given function $f \in \{f_\cup, f_\cup^T, f_\Sigma, f_\Sigma^T, f_\#\}$, 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* $f(M^{(0)}) = f(M^{(1)})$.

Table 1 defines the different $R_{\mathbf{N}}$ relations by the R_f relations.

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

B.1 DC-Net

The dining-cryptographers network protocol [11], is a multi party computation protocol. The protocol is based on David 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 publishes 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. 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 used. The protocol can be extended to n peers in different topologies, the most common is the ring.

B.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 participants 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).

$\mathbf{N}$	Notion	Definition of $R_{\mathbf{N}}$		
SUL	Sender Unlinkability	R_{SUL}	$\overset{def}{=}$	$R_{f_\Sigma} \cap R_{f_\cup^T}$
RUL	Receiver Unlinkability	R_{RUL}	$\overset{def}{=}$	$R_{f_\cup} \cap R_{f_\Sigma^T}$
UL	Unlinkability	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_Σ}
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 [21] table for anonymity variants defines each variant $\mathbf{N}$ and its associated relation $R_{\mathbf{N}}$

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. 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.

B.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 [4], DC-net is Comp-$\widehat{\mathbf{SA}}$-anonymous by the malicious destination extension of Section 6, 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 [16, Appendix A]. In a ring topology (of more than three peers), for breaking the anonymity of some peer, there is a need in both the peers on its sides [36].

Following the above claim, and if the attacker cannot break the encryption scheme, it is enough to prove that for every two scenarios with the same unprotected data, in every round the same messages are sent in both the scenarios.

Scenarios with the same unprotected data are defined by two matrices sequences $\{M^{(0)}_i\}^s_{i=1}$ and $\{M^{(1)}_i\}^s_{i=1}$, such that for every $1 \le i \le s$, $(M^{(0)}_i, M^{(1)}_i) \in R^{H,\tau}_{\mathbf{SA}}$ for some permutation τ over H's elements. 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 can be done by reducing the Comp-$\widehat{\mathbf{SA}}$-anonymity to the security of the encryption scheme.

C. CONTROLLING THE APPLICATION ADAPTIVELY MATTER

We now present a toy example of a protocol that ensures unobservability by [21] due to the inability of the adversary in their experiment to manipulate the application level adaptively, but not according to our definition:

The toy protocol for sending one message, is only for two participants, Alice and Bob. The protocol's run takes four iterations:

1. Alice sends a constant length message, m, encrypted by a CPA-Secure encryption scheme [4]. The message comes from the application, or is a dummy if the application has nothing to send.
2. Bob sends Alice a plain text with a random identifier $\in \{0,1\}^k$
3. Alice can send another message to Bob.
4. If Bob got the random identifier from Alice in the third iteration, he sends her m as a plain-text.

A passive global eavesdropper attacker that have non-negligible advantage according to definition (3), just inserts different messages into Alice's application queue in each scenario. After inspecting the random identifier that Bob sent in the second iteration, he adds the message identifier to Alice's application queue in both the scenarios, and guesses (with success probability 1) the simulated scenario by comparing Bob's response to the first message that Alice sent in each scenario.

But, according to [21]'s definition, the above toy protocol achieves unobservability. We shortly bring a proof sketch: For knowing the which scenario was simulated, the attacker has two options: Either to break the IND-CPA security of the encryption scheme, or to guess the random identifier (it has only one guess) with negligible success probability. Therefore, a simple reduction from the CPA-security of the protocol's encryption scheme to the **UO**-anonymity of the toy protocol will do the work.

D. EXAMPLE: TOR IS NOT COMP-N-ANONYMOUS AGAINST LOCAL EAVESDROPPER FOR ANY ANONYMITY NOTION N

In this appendix, we bring an example to anonymity protocol that its anonymity notions relies mainly on a lot of traffic, and show that it is not Comp-**N**-anonymous for any anonymity notion **N** (for some anonymity notion **N** [21]). We examine a simplified version of the Tor protocol [12], denoted by *Simp-Tor*.

In Simp-Tor there are three types of participants: clients, routers, and external sites. Every N iterations, every client chooses uniformly three different routers. A client who wants to send messages to sites (that are not necessary Tor clients), use the chosen three routers as a path, such that the last router sends the encrypted (even with perfect encryption scheme) messages to their destinations. The protocol is described in Algorithms 3 and 4.

Algorithm 3 *IterationAction*(iteration t). Client's iteration action in Simp-Tor.

```
if $t \bmod N = 0$ then
    $c_1, c_2, c_3 \leftarrow$ Choose 3 random routers
end if
$\{dest_i, m_i\}_{i=1}^{l} \leftarrow$ All the application messages.
Send to $c_1$: $\mathcal{E}_{PK_{c_1}}(c_2, \mathcal{E}_{PK_{c_2}}(c_3, \mathcal{E}_{PK_{c_3}}(\{dest_i, \mathcal{E}_{PK_{dest_i}}(m_i)\}_{i=1}^{l})))$
```

Algorithm 4 Router c iteration action in Simp-Tor

```
for all $\mathcal{E}_{PK_c}(NextDest, m)$ do
    Send $m$ to $NextDest$.
end for
```

For testing the protocol against local eavesdropper to one of the routers, we choose the next parameters for the experiment: π is the protocol. π's setup procedure returns a sequence of n states such that the first two states are of two Tor clients, the last two states are of two external sites (only receive messages and do not participate in the protocol), and the rest of the states are of Tor routers. π's setup initialize the states such that all the protocol participants know the relevant other identities and cryptographic keys. We only limit the n parameter such that $n \geq 7$ (there are at least 3 Tor routers). $\mathcal{A}$ is the $\mathcal{PPT}$ attacker, and its capability is $Cap = (\emptyset, \{3\})$. I.e., the attacker eavesdrops only to one participant - the one with $STATE_3$, which is one of the Tor routers.

We now present an attacker, $\mathcal{A}$, that for any **N**-anonymity notion [21] have non-negligible advantage according to definition (3); i.e., for every negligible function *negl*:

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

$\mathcal{A}$ works as follows:

1. Choose a random message $m' \in V$. Choose $M^{(0)}$ to be empty, except $m^{(0)}_{1,n-1} = m^{(0)}_{2,n} = \{m', m'\}$, and $M^{(1)}$ to be also empty, except $m^{(1)}_{1,n-1} = m^{(1)}_{1,n} = m^{(1)}_{2,n-1} = m^{(1)}_{2,n} = \{m'\}$.
2. In all the other iterations, choose $M^{(0)}$ and $M^{(1)}$ to be empty.
3. After the end of the fourth iteration, when the messages already reached their destinations, if the eavesdropped router sent 2 messages and to the same site, return 0. if it sent only 2 messages, but to different sites, return 1. Otherwise return random bit.

We now prove that the above $\mathcal{A}$ breaks the definition for every **N**-anonymity. First of all, all the pairs of matrices that were chosen by the attacker are in the $R^H = R^H_{f_\cup} \cap R^H_{f^T_\cup}$ relation, that contains $R^H_{\mathbf{N}}$ for each **N**.

We denote the number of Tor routers in the setup by c. $c = n - 4$ is polynomial in the security parameter k. $\mathcal{A}$ wins for sure, only if exactly one Tor client chose the eavesdropped router to be the last in his path. As the routers in the paths are chosen uniformly, the probability for such an event is $2 \cdot (\frac{1}{c} \cdot \frac{c-1}{c})$ for $c \geq 3$. In all the other cases $\mathcal{A}$ guesses, so its probability to win is $\frac{1}{2}$. therefore:

$$Pr[Expt^{Comp-\mathbf{N}-0}_{Simp-Tor,n,\mathcal{A},Cap}(k) = 1] = \frac{1}{2} \cdot (1 - 2 \cdot \frac{c-1}{c^2})$$

$$Pr[Expt^{Comp-\mathbf{N}-1}_{Simp-Tor,n,\mathcal{A},Cap}(k) = 1] = \frac{1}{2} \cdot (1 - 2 \cdot \frac{c-1}{c^2}) + 2 \cdot \frac{c-1}{c^2})$$

Therefore, for every negligible function, *negl*, the advantage of $\mathcal{A}$ is:

$$Adv^{Comp-\mathbf{N}}_{Simp-Tor,n,\mathcal{A},Cap}(k) = 2 \cdot \frac{c-1}{c^2}) \geq negl(k)$$

The last inequality is because $\frac{c-1}{c^2} \in poly(k)^{-1}$ which is not negligible in k. The same attack could be done when the attacker controls one of the routers. In similar algorithm that chooses routers path per destination, a similar attack will work; in this attack, the attacker wins if the eavesdropped or compromised router is chosen to be the first in one of the paths.

E. EFFICIENT RELAXED ULTIMATE ANONYMITY AND t-LIVENESS

We present the DPP (DPP), that ensures relaxed anonymity against strong malicious destination that controls $t < \sqrt{n}$ of the participants. DPP also satisfies t-liveness for $t < \sqrt{n}$.

For simplicity, we discuss a scheme where n participants need to send anonymous messages to one (malicious) destination, d.

The main problem of mixnet based protocol is malicious peers who drops messages of specific honest peers; then the

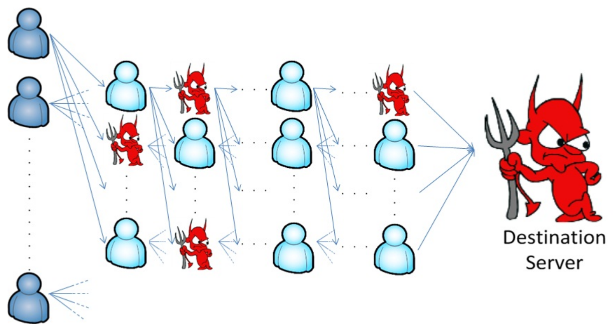

Figure 3: DPP's routing.

destination can distinguish between the scenarios by the messages it receives. Duplication of messages or using secret sharing [31] might not help, as the destination or the peers it controls, can still distinguish between the scenarios by the number of duplicated messages or the number of message's shares that reach the destination.

DPP overcome dropping or changing of messages by malicious participants by using $t+1$ disjoint paths to every relay. To overcome t malicious peers, DPP's routing uses $(t+1)^2$ peers as relays. The relays are sorted into $t+1$ levels of $t+1$ relays each. We denote the levels by l_i, and the j-th relay in l_i by $r_{i,j}$.

We now describe DPP's routing in high level: every round, all the potential senders send one message (real or dummy) to every relay in the first level (l_1). To deal with messages dropping or changing, in every l_i, $1 \le i \le t+1$ every relay collects all the messages it received (one copy for each message), and sends them randomly permuted, to every relay in l_{i+1}. The relays in l_{t+1} send the messages to the destination. Figure 3 depicts the routing procedure.

To prevent learning of the messages content (dummy or real, and any other information) and to enforce the routing, and prevent a case where malicious peer directly sends the message to the destination, the messages are encrypted with semantically secure encryption scheme [4], such that like in the Onion Routing protocol [27], the path of a message must be followed in order to get the content.

Unlike onion-routing, DPP has different $(t+1)^{t+1}$ paths for each message. Consequently, creating onions with each relay's public key is impossible. Therefore one common public key (and appropriate private key) is shared among the relays in the same level: (pk_i, sk_i) are the keys of the relays in l_i, and the onion of each message is done by encrypting the message with the destination's public key, and then with all pk_is from $i = t+1$ to 1.

The protocol can work also with symmetric encryption scheme (more efficient), assuming the senders knows all the keys, and every relay knows only its level key.

Every layer contains also the round where the message should be sent. That is to prevent reply attack: malicious relays in l_1 can send twice (in two different iterations) the same onion it got from some peer, and when the destination gets two identical messages, the originator will be exposed.

E.1 DPP Satisfies Relaxed Ultimate Anonymity and t-Liveness

We examine the relays as a $t+1 \times t+1$ matrix M, such that $M_{i,j} = r_{i,j}$. We denote an encrypted onion by O_0.

Algorithm 5 DPP's senders pseudocode. In the tuple (C, R, D), C is the message (possible ciphertext) to forward, R is the round to forward the C, and D is the next destination (the level of the next relays or the final destination).

```
    // $round$ is the current round
 1: $(m, d) \leftarrow$ Application.GetMessage()
 2: if $m = null$ then
 3:    $m \leftarrow dummy$
 4: end if
 5: $(C, R, D) \leftarrow (\mathcal{E}_{pk_d}(m), round + t + 1, d)$
 6: for $i = t + 1$ to 1 do
 7:    $(C, R, D) \leftarrow (\mathcal{E}_{pk_i}(C, R, D), r + i - 1, i)$
 8: end for
 9: for $j = 1$ to $t + 1$ do
10:    send $C$ to $r_{1,j}$
11: end for
```

Algorithm 6 DPP's pseudocode for the relay $r_{i,j}$.
Input: $t+1$ vectors of l shuffled encrypted messages. The relays in the first level (l_1) receives only l messages (referred as one vector).
Output: Vector of all the input messages without the outer encryption layer, to the next hop of the messages. The relays in l_{t+1} send the messages to their destinations.

```
    // $round$ is the current round
 1: $Set.init()$ // Initialize empty set of messages
 2: for all messages-vector $v$ in the input do
 3:    for $k = 1$ to $|v|$ do
 4:       $(C, R, D) \leftarrow \mathcal{D}_{sk_i}(v[k])$
 5:       if $round = R$ and $(D = i + 1$ or $i = t + 1)$ then
 6:          $Set.add(C, D)$
 7:       end if
 8:    end for
 9: end for
10: if $i < t + 1$ then
11:    $Vector.init(Set)$ // Create messages-vector from $Set$
12:    $Vector.shuffle()$ // Shuffle the messages in $Vector$
13:    for $k = 1$ to $t + 1$ do
14:       send $Vector$ to $r_{i+1,k}$
15:    end for
16: else
17:    for all message $(C, D)$ in $Set$ do
18:       Send $C$ to $D$
19:    end for
20: end if
```

We denote the partially peeled onion, after the relays in l_i decrypted their layer by O_i.

E.1.1 *t-liveness*

Every honest relay receives all the real messages (as onions). That is, because every M's row contains $t+1$ relays, and at most t of them are malicious, so there is a honest relay in every row. The honest relays in l_1 send all the real messages to all the relays in l_2, and from that point, every honest relay in l_i sends all the messages he received (including all the real messages, and maybe with additional messages added by malicious relays) to the relays in l_{i+1}. Therefore, in l_{t+1} there is at least one honest relay that receives all the messages and forward them to their destinations.

E.1.2 *Comp-$\widehat{\mathbf{SA}}$-Anonymity*

Due to the $R^{H,\tau}_{\mathbf{SA}}$ relation, in every round the same messages are sent, such that the only difference is in the senders. Therefore for breaking the anonymity, an attacker must connect or disprove connection between a message and its sender.

We relies on the following claim: Given an CPA secure encrypted message, it is computationally hard to learn about the message content [4].

Consequently, given two randomly shuffled sets of onions: $S_i = \{O_i^1, O_i^2, ..., O_i^p\}$ and $S_{i+1}\{O_{i+1}^1, O_{i+1}^2, ..., O_{i+1}^p\}$, such that the encryptions of the layers were done with the same keys for all the onions (onions of honest senders), the probability of an attacker to connect between some O_i^j to O_{i+1}^{j+1} is negligible. Otherwise, by a simple reduction, the encryption scheme is not CPA secure.

Because the number of malicious relays is less then $t+1$, in M there is at least one level without malicious relays. We denote the lowest index of such a level with h. From the routing procedure (see also the t-liveness proof), when the relays in l_h receives the messages, every one of them has all the real messages. From the security of the encryption scheme, from that point the attacker cannot link the onions. Every relay in l_h sends all the real messages shuffled (in one copy) to l_{h+1} such that from that point, the originators of the messages cannot be linked to the messages. Similarly to cascade mixnet [10].

E.1.3 *Comp-$\mathbf{UO}$-Anonymity*

As described above, DPP satisfies unobservability only if there is only one destination. Otherwise, an attacker can distinguish between two scenarios by destinations of the messages. To deal with many destinations, there is a need to remove the final destination from the message, and that the relays in l_{t+1} will send all the messages to all the possible destinations. In such a case, the encryption scheme should be also IK-CPA [2]. Namely, the attacker cannot distinguish between two messages that were encrypted by different keys. The unobservability relies only on the constant sending rate and the IND-CPA and the IK-CPA security of the encryption scheme. We omit here the reduction from the Comp-$\mathbf{UO}$-anonymity to the CPA-security of the encryption scheme.

E.2 Communication Complexity

Every message is encrypted to an onion. We analyze the communication complexity in the number of times an onion is sent. Every onion is sent once to each of the $t+1$ relays in l_1, and $t+1$ times to each relay in the other levels. The total number of 'send' events is therefore $t+1+t\cdot t\cdot(t+1) \in O(t^3)$. For every $t < \sqrt{n}$, the communication overhead is therefore $O(n^{1.5})$.