An Indistinguishability-based Characterization of Anonymous Channels

Alejandro Hevia* Daniele Micciancio†

July 20, 2008

Abstract

We revisit the problem of *anonymous communication*, in which users wish to send messages to each other without revealing their identities. We propose a novel framework to organize and compare anonymity definitions. In this framework, we present simple and practical definitions for anonymous channels in the context of computational indistinguishability. The notions seem to capture the intuitive properties of several types of anonymous channels (Pfitzmann and Köhntopp 2001) (eg. sender anonymity and unlinkability). We justify these notions by showing they naturally capture practical scenarios where information is unavoidably leaked in the system. Then, we compare the notions and we show they form a natural hierarchy for which we exhibit non-trivial implications. In particular, we show how to implement stronger notions from weaker ones using cryptography and dummy traffic – in a provably optimal way. With these tools, we revisit the security of previous anonymous channels protocols, in particular constructions based on broadcast networks (Blaze et al. 2003), anonymous broadcast (Chaum 1981), and mix networks (Groth 2003, Nguyen et al. 2004). Our results give generic, optimal constructions to transform known protocols into new ones that achieve the strongest notions of anonymity.

Keywords: Anonymous Channels, Computational Indistinguishability, Provable Security.

*Dept. of Computer Science, University of Chile, Blanco Encalada 2120, tercer piso, Santiago 837-0459, Chile. E-mail: `ahevia@dcc.uchile.cl` URL: `http://www.dcc.uchile.cl/ahevia`. Work partially done while the first author was at U. of California San Diego. Supported in part by Conicyt via Fondecyt grant No. 1070332.

†Dept. of Computer Science & Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-mail: `daniele@cs.ucsd.edu`, URL: `http://www-cse.ucsd.edu/users/daniele`. Research supported in part by NSF under grant CNS-0430595. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Contents

1 Introduction

Anonymous channels allow users to send and receive messages without revealing their identities. There are many applications for such channels, from protecting "whistle blowers" or guaranteeing source confidentiality in crime tips, to offering access to medical information to potential patients without fear of embarrassment, or protecting voter privacy in electronic voting [23, 43]. Chaum [14] initiated the modern study of anonymous communication by introducing the concept of mix networks (or *mix-nets*). A mix-net is a protocol in which messages (say, emails) traverse several routers (or mixers) and, in the process, are "mixed" with other messages with the intention that the relation to the original sender be lost. Since Chaum's seminal paper, research in the area has been extensive, from concrete mix-net proposals (see [47, 1, 39, 25, 33, 59] among many others) to very practical protocols based on mix-nets (eg. [29, 34, 40, 17, 51, 19] and references therein). But mix-nets are not the only method to implement anonymous communication. DC-nets (also known as anonymous broadcast networks), also proposed also by Chaum [15] and later improved by many others [10, 57, 58, 32], allow broadcast of messages without disclosing the sender identity. At least initially, most of the effort was put into improving the efficiency and reliability of the constructions, so informal or ad-hoc definitions were common. Indeed, only recently the need for general (and sound) definitions for these types of primitives has drawn some attention. Furukawa [24] and Nguyen et al. [44], in particular, give strong definitions for "proving shuffles" (shuffles are the basic mixing operation) and Wikström [59] presents a formal definition of mix-net in the UC model [13]. These definitions, although helpful in the design and analysis of mix-nets, do not provide a definition of anonymous channels per se. Indeed, the absence of good anonymity definitions that capture realistic concerns motivated this work.

OUR CONTRIBUTIONS: We present a novel framework to organize and compare anonymity definitions. In this framework, we formalize the notions of unlinkability, sender-anonymity, receiver-anonymity, sender-receiver anonymity, and unobservability, giving them new, strong indistinguishability-based formulations without compromising the standard "intuitive" meaning they have in the literature [46]. We also introduce new notions, namely sender unlinkability and receiver unlinkability. These notions, while arguably weak, can be used to implement some of the stronger notions. Then we formally prove some folklore results: we show that sender-receiver anonymity implies both sender anonymity and receiver anonymity, that sender-anonymity and receiver-anonymity (both separately) imply unlinkability, and that unobservability implies all the other properties. In the other direction, we present generic black-box transformations from any "weak" anonymous protocols (eg. sender unlinkability, unlinkability, or sender anonymity) into protocols anonymous under "stronger" notions (like sender-receiver anonymity or unobservability). These transformations are provably optimal in terms of message traffic. We then revisit the anonymity of constructions based on broadcast channels, DC-nets and mix-networks, giving an exact characterization of the anonymity they provide in our framework.

1.1 Coping with Information Leaks

There have been several attempts to characterize the intuitive properties anonymous channels should have. Most proposals so far seem to fall into two categories: (a) they present intuitive but weak definitions (targeted to particular applications with efficiency in mind), or (b) they present strong definitions with often impractical implementations [6, 28, 16]. We seek to bridge this gap by providing strong definitions which can be tailored to specific practical scenarios.

We identify factors or conditions that may realistically *limit* anonymity. These conditions are on specific information that, in principle, may be unrealistic to assume hidden from the adversary. Consider for example,

Anonymity Variant	Mnemonic Notation
Sender Unlinkability	(Σ, U)
Receiver Unlinkability	(U, Σ)
Sender-Receiver Unlinkability	(Σ, Σ)
Sender Anonymity	$(?, \mathsf{U})$
Receiver Anonymity	$(\mathsf{U}, ?)$
Strong Sender Anonymity	$(?, \Sigma)$
Strong Receiver Anonymity	$(\Sigma, ?)$
Sender and Receiver Anonymity	$(\#, \#)$
Unobservability	$(?, ?)$

Table 1: Anonymity variants and their associated mnemonic notation. The notation (X, Y) encodes what information is not assumed to be protected by the definition (ie. the meaning of X and Y), and from whom the information comes: from each sender (X), or each receiver (Y). 'U' stands for "`values of the messages sent/received`", 'Σ' for "`number of messages sent/received`", '$\#$' for "`total number of messages`", and '?' for "`nothing`".

(a) **Total network flow is usually public:** the total number of messages sent in a system is likely to be known to any party in the system, even external observers.

(b) **Amount of traffic per party is hard to conceal:** the number of messages sent or received by a particular party is often easily inferred by an observer in the party's network vicinity.

(c) **Values sent or received by each party are not necessarily private:** the value of each message[1] sent or received by a particular party could be guessed, known, or even influenced by an adversary.

A proper definition of anonymity should take these "leaks" into account but hide any additional information: *hide everything except what follows from the potentially leaked information*. This idea is already present in security definitions of other cryptographic primitives. For example, if E is a semantically secure encryption function [30], it is standard to assume a ciphertext $E(m)$ hides all partial information about a message m except its length $|m|$. This is because $|m|$ can only be hidden at the cost of unnecessarily increasing the size of $E(m)$. In fact, the definitions in this work are inspired by the indistinguishability-based formalization of semantically secure encryption in [30], which guarantees the hiding of all information on the plaintext other than the plaintext length. Similarly, an anonymous channel should hide all information about the communication except for (some of) the information mentioned above. In this work, we study the possible combinations of the conditions (a),(b), and (c) above, and analyze the resulting notions. There are nine (potentially different) notions. Named following the intuition in [46], they are summarized in Table 1.

Toy examples of the traffic patterns protected by all variants are shown in Appendix B. *Sender Unlinkability* and *Receiver Unlinkability* are the weakest notions of anonymity we consider. A protocol is sender unlinkable if it hides any relation between senders and receivers beyond what is implied by the total size of messages sent by each party and the specific values of the messages received by each party. Its dual notion is *Receiver Unlinkability* in which the roles of sender and receiver are reversed. Compared to Receiver Unlinkability, *Sender and Receiver Unlinkability* (or simply *Unlinkability*) strengthens the requirements for the sender, hiding the message values sent and received but not necessarily the total size of messages exchanged by each party. A stronger notion is *Sender Anonymity* as the number and values of messages for the

[1] We distinguish two properties for each message: its value, that is, the data or *payload* encoded in the message, and its destination.

sender must remain hidden (but not the values of the received messages for each party). Compared to Sender Anonymity, *Receiver Anonymity* simply reverses the roles of sender and receiver. Further strengthening of these notions are *Strong Sender Anonymity* (resp. *Strong Receiver Anonymity*) in that protocols can afford to leak at most the amount of traffic per receiver (resp. per sender). The strongest notions are *Sender-Receiver Anonymity*, and *Unobservability*. They differ in that the former may not protect the total network flow (ie. the total number of messages exchanged), while the latter must hide this information.

1.2 Strong, Formal Definitions

We adopt an indistinguishability based formalization under which the adversary produces two message matrices (which encode message senders and receivers in a standard way), is allowed to passively observe the execution of a communication protocol under a random one of these two matrices and then is required to have non-negligible advantage in determining under which of the two matrices the protocol was executed. Within this framework, each different anonymity variant is defined by requiring the adversary to produce two matrices whose "leaked" information is the same. More precisely, if for any message matrix M the anonymity variant assumes a certain information $f(M)$ may not be protected (it may be "leaked"), then the two matrices M, M' produced by the adversary must satisfy $f(M) = f(M')$. Indeed, the notions corresponding to the different anonymity variants mentioned in the previous section follow from instantiating function f with the appropriate function (eg. one that computes the set of message values sent per party, their number, or the total number of messages, for example). Our formalisms build on definitional ideas used for encryption [30, 42, 27] and signatures [31]. Regarding adversaries, an often adopted adversarial type is that of *honest-but-curious* (or passive) adversary, one where the adversary obtains the internal state of the corrupted party, but the party continues to follow the protocol. For simplicity of exposition, we consider passive adversaries with no corruptions (also called *outside* [20] or *global passive adversary* [52]) as it captures most of the subtleties of our model. Extensions to allow (passive) corruptions are discussed in Section 6. We also stress that our results apply to protocols with fixed number of participants.

Since the adversary can freely choose the values and destinations of all messages in the protocol (ie. the message matrix), it follows that a protocol anonymous under this definition must hide all partial information on the message matrix M *except for what is implied by the known information* $f(M)$. In particular, sources and destinations of the messages are hidden up to the extent that they do not follow from the known information. This is a quite strong guarantee.

We stress that we present an unified framework for *all the proposed anonymity variants*. We believe this facilitates the organization and comparison of the notions as well as future extensions.

1.3 Comparing Notions

The indistinguishability-based definitions presented in this paper appear to capture the concerns of most intuitive but informal notions of anonymity proposed in the past [46]. Indeed, in Section 1.4 we argue that previous anonymity formalizations in comparable network models are implied by some of the proposed notions. In addition, we compare the new notions to each other. The comparison is in terms of reductions. We say notion A implies (is stronger than) notion B if any protocol satisfying A can be used to achieve B (via a possibly different protocol). A difficulty arises if we assume point-to-point channels between parties. In this case, protocols for all notions exist because of general secure multiparty computation results [6, 28, 16], which makes the notions trivially equivalent. To avoid this pitfall, we assume that the only communication channel between the parties is an idealized version of a protocol achieving notion A, and then we show how to implement a protocol that achieves notion B in this setting. The communication channel is idealized in the sense that parties only see its input/output behavior. This effectively gives us black-box reductions.

RESULTS: We show three types of reductions between the anonymity definitions: (1) Trivial reductions, in which given a protocol for notion A, the same protocol achieves notion B, (2) Reductions that use cryptography, and (3) Reductions that use "padding" (or "dummy traffic"). Interestingly, in terms of the reductions, cryptography and padding do not appear exchangeable. Our results suggest that in the reductions that require cryptography padding does not help, while in those where padding is necessary, cryptography does not help.

TRIVIAL REDUCTIONS: There exists a partial order of the notions, starting from the weakest ones, sender unlinkability and receiver unlinkability, and ending in the strongest one, unobservability, such that if a protocol achieves a certain notion then the same protocol achieves any weaker notion. These relations give formal justification to previous informal statements such as sender-receiver anonymity implying both sender anonymity and receiver anonymity, or that unobservability implies all the other notions. Interestingly, there is no trivial relation between sender anonymity, unlinkability, and receiver anonymity, which indicates the definitions address incomparable security concerns. In [46], however, it is argued that Unlinkability (called "relationship anonymity" there) is a "weaker property than each of sender anonymity and recipient anonymity". The disagreement disappears when one notices that, under our definitions, such relation is true between *strong* sender (or receiver) anonymity and unlinkability. Our framework allows us then to clarify an implicit assumption in [46], namely that messages in the definitions of sender and receiver anonymity are private.

USING CRYPTOGRAPHY: Under standard computational and setup assumptions, we show that anonymity notions that reveal message values are not intrinsically weaker than those that keep these values private. In particular, we show reductions from unlinkability to sender (or receiver) unlinkability. We also show that strong sender (resp. receiver) anonymity is not weaker than sender (resp. receiver) anonymity.[2] The assumptions are standard, namely PKI and key-private secure encryption schemes [4].[3] The reductions are computationally efficient and do not have message overhead – they introduce no new messages – therefore optimal in terms of communication.

USING "PADDING": We conclude showing that our strongest anonymity notions *can* be achieved starting from much weaker anonymity notions, but at a cost of message efficiency. In a nutshell, the reductions show that unobservability, sender-receiver anonymity, strong sender (or receiver) anonymity, and unlinkability are actually equivalent. They also show that neither sender nor receiver unlinkability are stronger than sender or receiver anonymity. These reductions do introduce *dummy traffic* (ie. extra empty messages) but no more than necessary – they have optimal message overhead. These reductions do not require computational or setup assumptions, and are computationally efficient.[4] The results are summarized in Fig. 2.

1.4 Comparison with Previous Anonymity Notions

In this section, we compare the proposed variants with anonymity variants suggested previously in the literature. When necessary, we relax those definitions to match our adversarial model (passive adversaries with no corruptions).

[2] This proof actually *justifies* the assumption made in [46] mentioned before. We stress that this is not obvious since anonymity does not necessarily implies message privacy, or viceversa.

[3] In fact, based on preliminary results, we conjecture computational or setup assumptions are also necessary.

[4] The reductions *to* Sender Anonymity, Strong Sender Anonymity, and Unobservability require the extra (but rather mild) assumption that a known upper bound on the total network flow exists. See Proposition 4.6 and remarks at the end of Section 4.2.

INDISTINGUISHABILITY-BASED DEFINITIONS: Beimel and Dolev [3] define anonymity in terms of computational indistinguishability of the adversary's *view* (i.e. the messages and any extra information obtained by the adversary) in two cases: when party P_i sends a message to party P_j, and when $P_{i'}$ sends a message to $P_{j'}$, for any i, j, i', j'. Given that [3] does present protocols for multiple senders, we see the definition as somewhat unsatisfactory in the following sense. The definition does not specify how the messages and destinations for parties $P_k \neq P_i$ are selected. If they are chosen either arbitrarily (but the same for both views) or with some probability distribution, then we can show they are strictly *weaker* than sender-receiver anonymity. The alternative, choosing the inputs for parties $P_k \neq P_i$, arbitrarily but different in each view, might work (be equivalent to sender-receiver anonymity) although it is unclear without a formal statement. A similar concern can be raised on the definition proposed by von Ahn et al. in the context of k-anonymity [56]. (Essentially the same definition for the case of a fixed receiver).
Golle and Juels [32] present a definition of anonymity (which they called privacy) in the context of DC-nets [15]. In the definition in [32], a successful adversary must distinguish between an execution where P_1 sends a message to some party P_b, and one in which P_2 sends a message to some party P_{1-b}, where b is a bit chosen uniformly at random and *unknown* to the adversary. The rest of the parties sends messages as instructed by the adversary. Unfortunately, this definition suffers from a problem similar to the one above. The adversary is unable to exploit possible correlations between the destination of P_1's message and the destination of some other party P_3's message. Consequently, this definition can be shown to be strictly weaker than our definition of sender anonymity. Luckily, the DC-net in [32] is strong enough to be proven sender anonymous (see Section 5.2).

OTHER CLOSELY RELATED DEFINITIONS: Nguyen et al. [44] define privacy of a shuffle by a similar experiment to ours (a notion called indistinguishability under chosen permutation attack or IND-CPA$_S$ under an active adversary). In their definition, the adversary chooses two permutations under which the messages are shuffled and must distinguish which one was used. Translated to our setting, their definition restricts message matrices to be permutations such that each party sends exactly a single message. Also, it does not account for the types of information leaks we consider. The comparison is somewhat unfair, as their concern – privacy of a single shuffle – is different than ours.

Another related definition was suggested (rather implicitly) by Ishai et al. in [38]. There, Ishai et al. describe a functionality for anonymous communication (synchronous setting with rushing). When paired with the appropriate notions of multiparty computation [12] (under our adversarial model), their definition becomes a special case of ours, namely Sender Anonymity (SA). Their work [38], however, does not explore the proposed definition but instead use it to prove the security of other (non-anonymity related) cryptographic protocols.

Recently and independently from our work, Feigenbaum et al. [22] presented a definition of anonymity which, although it was specially tailored to the onion-routing system Tor [19], is closed to ours in spirit. In their work, several variants of anonymity are defined in terms of indistinguishability of configurations, where configurations may include values and destination of messages sent by parties in the system. When considered under our adversarial model, their definition differs from ours as there the indistinguishability property is explicitly expressed in terms of *circuits* (a routing path of a given message sent in any onion-routing system) and messages/actions on them, while our definition does not assume onion-routing-type of operation nor any particular underlying communication system. And, while our definition does seem to capture a wider variety of anonymity variants, the definition in [22] does allow an (arguably) stronger adversarial model. None of the definitions above incorporates provisions to deal with "leaked" information on the granularity done in the present work though.

1.5 Related Work

Dolev and Ostrovsky [20] present "xor-trees" protocols, a generalization of DC-net into a spanning tree, which they prove secure under a notion based on the concept of anonymity set (see below). Similarly, Pfitzmann [45] proposes the notion of k-anonymity – further developped by [56] – which can be seen as an extension of the DC-net model to more practical graph structures (which partition the parties into k-sized autonomous groups). Another approach was proposed by Rackoff and Simon in [49]. They describe a protocol for anonymous communication based on sorting networks, which is shown to satisfy some statistical mixing properties. Relaxations to weaker adversaries were proposed by Reiter and Rubin [50] and Berman et al. [7]. Both works presented alternative notions of anonymity as well as efficient constructions assuming an adversary that does not monitor all communication channels. Camenisch and Lysyanskaya [11] give a formal definition of onion routing [29] (along a provable secure protocol) but they explicitly avoid defining anonymous channels.

An alternative characterization of anonymity has been through the concept of anonymity set [15, 40]. The anonymity set is defined as the set of parties that could have sent a particular message as seen from the adversary [46]. Follow up works [40, 53, 18] have proposed new characterizations of anonymity, mostly in terms of the probability distributions the adversary assigns to each party in order to represent the likelihood such party is the sender of a message. Definitions based on formal methods have also been proposed [55, 37, 52, 41, 26]. Finally, it is worth noticing that Hughes and Shmatikov [36] also present a framework to formalize and compare different notions of anonymity as done here. Using the domain-theoretic primitive of function-view they model different notions of anonymity where information leaks can in principle be factored into the model. Their results, however, are not inmediately comparable to ours, as they focus only on non-probabilistic observers (adversaries) while ours can be probabilistic as long as they are efficiently computable.

ORGANIZATION: The rest of the paper is organized as follows. Section 2, introduces some notation and details on the execution model. Then, in Section 3, we present the formal definition of anonymous channels. Section 4 presents implications between the notions as well as proofs of their optimality in terms of communication. Then, in Section 5, we revisit previously proposed anonymous protocols and examine their security in the current framework. We conclude in Section 6 mentioning some extensions to the model.

2 Preliminaries

MODEL AND NOTATION: We consider a system of n parties $P_1, \ldots, P_n$, where n is polynomial in the security parameter $k \in \mathbb{N}$, connected to each other by point-to-point communication channels. We distinguish two (possibly overlapping) types of parties: senders and receivers. For any two finite sets A and B, let $A \uplus B$ denote the multiset union (also called sum or join) of A and B, and $|A|$ denote the size of multiset A. By convention, we assume the i, j-th element of any matrix $M = (m_{i,j})_{i,j \in [n]}$ is denoted by $m_{i,j}$. As usual, M^T denotes the transpose of any matrix M, and $m_{i,*} = (m_{i,j})_{j \in [n]}$ a matrix row.

MESSAGES: We let $V = \{0,1\}^\ell$ denote the message space where $\ell = \ell(k)$ for a polynomial $\ell(\cdot)$. The collection of messages sent by parties as well as their destinations is an $n \times n$ matrix $M = (m_{i,j})_{i,j \in [n]}$, called the *message matrix*. For row index i and column index j, $m_{i,j} \in \mathcal{P}(V)$ is the (multi)set of messages from party P_i to party P_j.[5] The *size* of matrix M, i.e. the total number of messages sent, is denoted by

[5] We abuse the notation and we see elements of $\mathcal{P}(V)$ as multisets. This extension is needed to consider parties that send duplicated messages to the same receiver (see Section 4.2).

$|M| \stackrel{\text{def}}{=} \sum_{i,j\in[n]} |m_{i,j}|$.

ADVERSARIES AND PROTOCOL EXECUTION: In our setting, adversaries are (possibly external) PPT parties in the system which can passively monitor all the communication between parties. We consider only *passive adversaries* that do not corrupt any party but are able to read (but not alter) all the messages exchanged by the parties. A protocol π is a sequence of instructions that all parties (senders and receivers) must follow. The instructions involve local computations and point-to-point message exchanges between parties. Our execution model is a special case of the model presented by Canetti [12] (since we consider only passive adversaries). Given a message matrix M, we define the execution of protocol π with input M under adversary A, as the process where each party P_i follows the instructions of protocol π using as input the i-th row $m_{i,*}$ of matrix M. In this process, we allow the adversary A to obtain a copy of all messages exchanged in all communication channels. We say protocol π is a *message-transmission protocol* if, for any PPT adversary A and any message matrix M, each receiver P_j's local output y_j after executing π on input M equals the multiset $\uplus_{i\in[n]} m_{i,j}$.

3 Security Notions

Our definition is formalized in an *indistinguishability-type experiment* following similar approaches used in the formalization of semantically secure encryption schemes [5]. We define anonymity via an *experiment* or *game*, in which there are two "worlds" (world 0 and world 1). We allow the adversary to choose the messages (values and destinations) sent by each party in each world. These choices are represented by two message matrices $M^{(0)}$ and $M^{(1)}$. Then, world $b \in \{0,1\}$ is chosen uniformly at random, and message-transmission protocol π is executed by all parties on input $M^{(b)}$. We measure the adversary's success in terms of her ability to distinguish the two worlds.

Our definition is inspired by the standard game used to define semantically secure encryption scheme, namely the *left-or-right* characterization of IND-CPA [5]. There, the adversary arbitrarily chooses two messages of the same length, is returned an encryption of a random one of the two messages and then is required to guess under which message the encryption was generated. The adversary's inability to distinguish the plaintext underlying in the ciphertext effectively means she cannot compute any information on the plaintext except its length [30, 5]. Similarly, the definition of our anonymity game guarantees that no information can be efficiently computed on the destinations of the messages sent during the protocol.

As mentioned in the introduction, one important difference between our formulation and the left-or-right game mentioned above is that we restrict the adversary's choices of the values and destinations of the messages to capture what is known to the adversary. These restrictions are captured as follows. Let f_{U}, f_{Σ}, and $f_{\#}$ be functions that map matrices $M = (m_{i,j})_{i,j\in[n]}$ into $\mathcal{P}(V)^n$, $\mathbb{N}^n$, and $\mathbb{N}$ respectively, defined by

$$\begin{aligned} f_{\mathsf{U}}(M) &\stackrel{\text{def}}{=} (\uplus_{j\in[n]} m_{i,j})_{i\in[n]}\,, \\ f_{\Sigma}(M) &\stackrel{\text{def}}{=} (\sum_{j\in[n]} |m_{i,j}|)_{i\in[n]} \quad \text{and} \\ f_{\#}(M) &\stackrel{\text{def}}{=} |M|\,. \end{aligned}$$

Also, let $f_{\mathsf{U}}^T(M) \stackrel{\text{def}}{=} f_{\mathsf{U}}(M^T)$, and $f_{\Sigma}^T(M) \stackrel{\text{def}}{=} f_{\Sigma}(M^T)$. Associated to each function f there is an equivalence relation $R_f \subset \mathcal{M}_{n\times n}(\mathcal{P}(V))^2$ where $(M, M') \in R_f$ if and only if $f(M) = f(M')$. For simplicity, we denote $R_{\mathsf{U}} = R_{f_{\mathsf{U}}}$, $R_{\mathsf{U}}^T = R_{f_{\mathsf{U}}^T}$, $R_{\Sigma} = R_{f_{\Sigma}}$,$R_{\Sigma}^T = R_{f_{\Sigma}^T}$, and $R_{\#} = R_{f_{\#}}$.

$\mathbf{N}$	Notion	Description of $R_{\mathbf{N}}$
SUL	Sender Unlinkability	$R_{\mathsf{SUL}} \stackrel{\text{def}}{=} R_{\Sigma} \cap R_{\mathsf{U}}^{T}$
RUL	Receiver Unlinkability	$R_{\mathsf{RUL}} \stackrel{\text{def}}{=} R_{\mathsf{U}} \cap R_{\Sigma}^{T}$
UL	Unlinkability	$R_{\mathsf{UL}} \stackrel{\text{def}}{=} R_{\Sigma} \cap R_{\Sigma}^{T}$
SA	Sender Anonymity	$R_{\mathsf{SA}} \stackrel{\text{def}}{=} R_{\mathsf{U}}^{T}$
RA	Receiver Anonymity	$R_{\mathsf{RA}} \stackrel{\text{def}}{=} R_{\mathsf{U}}$
SA^*	Strong Sender Anonymity	$R_{\mathsf{SA}^*} \stackrel{\text{def}}{=} R_{\Sigma}^{T}$
RA^*	Strong Receiver Anonymity	$R_{\mathsf{RA}^*} \stackrel{\text{def}}{=} R_{\Sigma}$
SRA	Sender-Receiver Anonymity	$R_{\mathsf{SRA}} \stackrel{\text{def}}{=} R_{\#}$
UO	Unobservability	$R_{\mathsf{UO}} \stackrel{\text{def}}{=} \mathcal{M}_{n\times n}(\mathcal{P}(V))^2$

Figure 1: Anonymity variants and their associated relations $R_{\mathbf{N}}$.

We are now ready to present the main definition. Given an n-party message-transmission protocol π, an adversary A, and label $\mathbf{N} \in \{\mathsf{SUL}, \mathsf{RUL}, \mathsf{UL}, \mathsf{SA}, \mathsf{RA}, \mathsf{SA}^*, \mathsf{RA}^*, \mathsf{SRA}, \mathsf{UO}\}$, consider the experiment $\mathbf{Exp}_{\pi,A}^{\mathbf{N}-anon}(k)$ described below. The experiment is parameterized by label $\mathbf{N}$, which determines the relation $R_{\mathbf{N}}$ considered. Relation $R_{\mathbf{N}}$ is defined in terms of R_{U},R_{U}^{T},R_{Σ},R_{Σ}^{T} and $R_{\#}$ according to the table in Fig. 1. We define the success probability of adversary A attacking protocol π under notion $\mathbf{N}$ as $\mathbf{Adv}_{\pi,A}^{\mathbf{N}-anon}(k) \stackrel{\text{def}}{=} 2 \cdot \Pr\left[\mathbf{Exp}_{\pi,A}^{\mathbf{N}-anon}(k) = 1\right] - 1$ where the experiment is defined as follows:

Experiment $\mathbf{Exp}_{\pi,A}^{\mathbf{N}-anon}(k)$
  $b \stackrel{R}{\leftarrow} \{0,1\}$, and $\langle M^{(0)}, M^{(1)}\rangle \leftarrow A(k)$
  if $\langle M^{(0)}, M^{(1)}\rangle \notin R_{\mathbf{N}}$ **then return** 0
  else Execute π on input $M^{(b)}$ under adversary A until A outputs a bit g.
    if $(b = g)$ **return** 1 **else return** 0

Definition 3.1 (Anonymous Channels) A message-transmission protocol π achieves $\mathbf{N}$-*anonymity* for $\mathbf{N} \in \{\mathsf{SUL}, \mathsf{RUL}, \mathsf{UL}, \mathsf{SA}, \mathsf{RA}, \mathsf{SA}^*, \mathsf{RA}^*, \mathsf{SRA}, \mathsf{UO}\}$, if for all PPT adversaries A, the quantity $\mathbf{Adv}_{\pi,A}^{\mathbf{N}-anon}(k)$ is negligible in $k \in \mathbb{N}$. ▌

4 Relation between the Notions

In this section, we show implications between the notions. We start by formalizing the type of reduction we use.

BLACK-BOX IMPLICATIONS: As mentioned before, we consider a simplified network where the only communication channel between the parties is an idealized implementation of a protocol satisfying a certain anonymity notion N_1. We say notion N_1 *implies* notion N_2 (or alternatively that N_2 *reduces to* N_1), denoted by $N_1 \rightarrow N_2$, if there exists a protocol $\theta^{(\cdot)}$ with access to the idealized communication channel such that, for every protocol π, the following holds: if π achieves N_1-anonymity, then θ^{π} achieves N_2-anonymity.

RESULTS: Our results are summarized in Fig. 2. We first describe some easy implications, most of them folklore results, which until now remained without formal proof. An interesting aspect of the result is that the transformation which enables the reductions is the identity function. Therefore, some definitions are stronger than others in the sense that any protocol achieving one definition also achieves the other one.

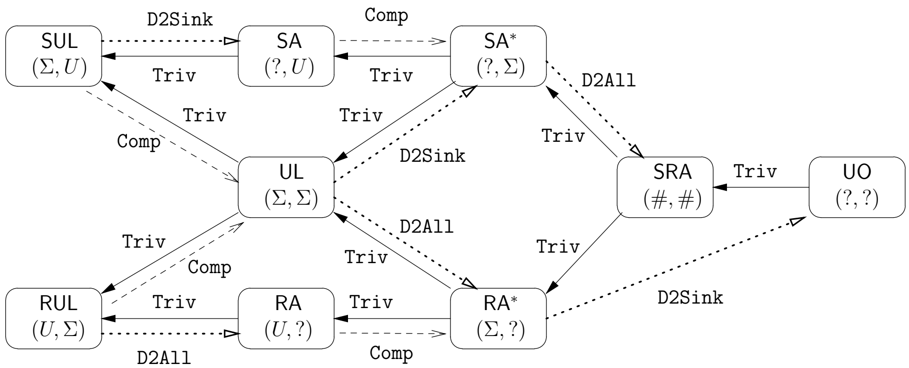

Figure 2: Relations among notions of anonymity. Arrows labeled `Triv` denote trivial implications (Proposition 4.1) and those labeled `Comp` denote implications under computational assumptions (Lemma 4.2). Arrows labeled `D2Sink` and `D2All` denote implications that use the transformation of the same name (Proposition 4.6 and Proposition 4.7 respectively). Implications obtained by transitivity are not drawn.

Proposition 4.1 The following implications hold unconditionally $\mathsf{UO} \rightarrow \mathsf{SRA} \rightarrow \mathsf{SA}^* \rightarrow \mathsf{SA} \rightarrow \mathsf{SUL}$, $\mathsf{SRA} \rightarrow \mathsf{RA}^* \rightarrow \mathsf{RA} \rightarrow \mathsf{RUL}$, $\mathsf{SA}^* \rightarrow \mathsf{UL} \rightarrow \mathsf{RUL}$ and $\mathsf{RA}^* \rightarrow \mathsf{UL} \rightarrow \mathsf{SUL}$. ▌

Proof of Proposition 4.1: First, we notice that, by definition $R_{\mathsf{U}} \subset R_{\Sigma} \subset R_{\#}$ and $R^T_{\mathsf{U}} \subset R^T_{\Sigma} \subset R_{\#}$. The results follows easily from these relations. We illustrate this by proving the implication $\mathsf{UL} \rightarrow \mathsf{SUL}$. The other implications are similar. In order to prove that $\mathsf{UL} \rightarrow \mathsf{SUL}$, it suffices to show that, for any protocol π, given a good SUL-adversary A, there exists a good UL-adversary A'. Since $R^T_{\mathsf{U}} \subset R^T_{\Sigma}$, then it follows that $R_{\mathsf{SUL}} \subset R_{\mathsf{UL}}$ and, in consequence, any SUL-adversary A for protocol π is also a UL-adversary for the same protocol, so taking $A' = A$ suffices. ▌

4.1 Implications under Computational Assumptions

In this section, we show that, under some standard setup and computational assumptions (namely PKI and key-private secure encryption [30, 4]), some of the notions are equivalent in the sense that a protocol achieving one definition can be efficiently transformed into a similar protocol achieving the other definition. In particular, RUL,SUL, and UL are all equivalent, as well as SA and SA*, and RA and RA*. The assumptions and their formalization are reviewed in Appendix A.

Lemma 4.2 Assume key-private semantically secure public-key encryption schemes and PKI exist. Then $\mathsf{SUL} \rightarrow \mathsf{UL}$, $\mathsf{RUL} \rightarrow \mathsf{UL}$, $\mathsf{SA} \rightarrow \mathsf{SA}^*$ and $\mathsf{RA} \rightarrow \mathsf{RA}^*$. ▌

For each implication of the lemma, the structure of the proof is the same and is divided into two steps. To prove that notion $\mathbf{N}$ implies notion $\mathbf{N}'$, we first define an intermediate notion, called $\mathbf{I}$-$\mathbf{N}$-*anonymity* (or *value oblivious* $\mathbf{N}$-*anonymity*, which we prove is implied by $\mathbf{N}$, that is, $\mathbf{N} \rightarrow \mathbf{I}$-$\mathbf{N}$. Then, we prove that $\mathbf{I}$-$\mathbf{N}\rightarrow \mathbf{N}'$. Interestingly, the proof that $\mathbf{N} \rightarrow \mathbf{I}$-$\mathbf{N}$ is the same for $\mathbf{N} \in \{\mathsf{SUL}, \mathsf{RUL}, \mathsf{SA}, \mathsf{RA}\}$, so we present it only once, first. The new notions, although somewhat technical, are the natural extensions of relations R_{U} and R^T_{U} to capture indistinguishability of the values instead of equality. Proving that the resulting notion $\mathbf{I}$-$\mathbf{N}$ is in fact *implied* by the original notion $\mathbf{N}$ is nonetheless non-trivial.

Let $\mathbf{N} \in \{\mathsf{SUL}, \mathsf{RUL}, \mathsf{SA}, \mathsf{RA}\}$. Given $\mathbf{N}$-anonymity, we define notion $\mathbf{I}$-$\mathbf{N}$-anonymity using an experiment similar to that underlying the definition of $\mathbf{N}$-anonymity. In fact, the only difference is that the adversary can specify two PPT *sampling* algorithms $G^{(0)}$ and $G^{(1)}$ from where the elements of the challenge matrices $M^{(0)}, M^{(1)}$ are drawn. The only restriction is that $G^{(0)}$ and $G^{(1)}$ must induce computationally indistinguishable ensembles.[6] Intuitively, this experiment decouples the adversary's control over message values and message destinations. Matrices $M^{(0)}, M^{(1)}$ specify the adversarial choices for sources and destinations of messages, while the sampling pair $(G^{(0)}, G^{(1)})$ specifies distributions for the message values. Details follow.

Let $k \in \mathbb{N}$ be a security parameter. For simplicity, assume that each party only sends a single message to each other party.[7] Two algorithms $G^{(0)}(\cdot,\cdot)$ and $G^{(1)}(\cdot,\cdot)$ form an *indistinguishable sampling pair* if each is PPT on the first input, and the ensembles $\left\{G^{(0)}(k,a)\right\}_{k\in\mathbb{N},a\in V}$ and $\left\{G^{(1)}(k,a)\right\}_{k\in\mathbb{N},a\in V}$ are computational indistinguishable. We say PPT algorithm A is a *legal* adversary if, on input k, A's first output is a tuple $(M^{(0)}, M^{(1)}, \langle G^{(0)}\rangle, \langle G^{(1)}\rangle)$ where $M^{(0)}, M^{(1)}$ are message matrices and $\langle G^{(0)}\rangle, \langle G^{(1)}\rangle$ is the encoding of an indistinguishable sampling pair. Given a legal adversary A, we define the experiment $\mathbf{Exp}^{\textbf{I-N}-anon}_{\pi,A}$ as described below. The corresponding success probability $\mathbf{Adv}^{\textbf{I-N}-anon}_{\pi,A}(k)$ of adversary A is defined in the usual way.

Experiment $\mathbf{Exp}^{\textbf{I-N}-anon}_{\pi,A}(k)$

$b \xleftarrow{R} \{0,1\}$, and $(M^{(0)}, M^{(1)}, \langle G^{(0)}\rangle, \langle G^{(1)}\rangle) \leftarrow A(k)$

if $(M^{(0)}, M^{(1)}) \notin R_{\mathbf{N}}$ **then return** 0

else Parse $M^{(0)}$ as $(m^{(0)}_{i,j})_{i,j\in[n]}$ and $M^{(1)}$ as $(m^{(1)}_{i,j})_{i,j\in[n]}$

For all $i,j \in [n]$, all $d = 0,1$,

if $m^{(d)}_{i,j} \neq \emptyset$, then set $\bar{m}^{(d)}_{i,j} \xleftarrow{R} G^{(d)}(k, m^{(d)}_{i,j})$, or $\bar{m}^{(d)}_{i,j} \leftarrow \emptyset$ otherwise.

$\bar{M}^{(0)} \leftarrow (\bar{m}^{(0)}_{i,j})_{i,j\in[n]}$ and $\bar{M}^{(1)} \leftarrow (\bar{m}^{(1)}_{i,j})_{i,j\in[n]}$

Execute π on input $\bar{M}^{(b)}$ under adversary A until A outputs a bit g.

if $(b = g)$ **return** 1 **else return** 0

For completeness, the formal definition is presented next.

Definition 4.3 Let $\mathbf{N} \in \{\mathsf{SUL}, \mathsf{RUL}, \mathsf{SA}, \mathsf{RA}\}$. A message-transmission protocol π achieves $\mathbf{I}$-$\mathbf{N}$-*anonymity* if for all legal PPT adversaries A, the quantity $\mathbf{Adv}^{\textbf{I-N}-anon}_{\pi,A}(k)$ is negligible in $k \in \mathbb{N}$. ▌

We obtain the result of the lemma from the following two propositions. The first one shows that $\mathbf{N} \to \mathbf{I}$-$\mathbf{N}$ for any notion $\mathbf{N} \in \{\mathsf{SUL}, \mathsf{RUL}, \mathsf{SA}, \mathsf{RA}\}$, and the second one proves the results of the lemma starting from $\mathbf{I}$-$\mathbf{N}$. Intuitively, this proposition states that the adversary's ability to *choose* the input values for the messages does not weaken the notion of anonymity.

Proposition 4.4 Let $\mathbf{N} \in \{\mathsf{SUL}, \mathsf{RUL}, \mathsf{SA}, \mathsf{RA}\}$, and let π be a message-transmission protocol that achieves $\mathbf{N}$-anonymity. Then, π achieves $\mathbf{I}$-$\mathbf{N}$-anonymity. ▌

Proof of Proposition 4.4: Fix $\mathbf{N} \in \{\mathsf{SUL}, \mathsf{RUL}, \mathsf{SA}, \mathsf{RA}\}$. In this case, it is easy to see that for any two message matrices $M^{(0)} = (m^{(0)}_{i,j})_{i,j\in[n]}$ and $M^{(1)} = (m^{(1)}_{i,j})_{i,j\in[n]}$ that belong to relation $R_{\mathbf{N}}$, there exist a

[6] At first look, this type of adversary may seem artificial, as the restrictions on the sampling algorithms cannot be efficiently tested. Nonetheless, this is all we need, as Proposition 4.5 shows that for each implication $\mathbf{I}$-$\mathbf{N} \to \mathbf{N}'$ any $\mathbf{N}'$-adversary can be transformed into this type of $\mathbf{I}$-$\mathbf{N}$-adversary, which in turn Proposition 4.4 shows can be mapped into an "regular" $\mathbf{N}$-adversary.

[7] The implications still hold if more than one message is exchanged between each pair of parties although the proof becomes a little more involved.

permutation $\rho\colon [n]^2 \to [n]^2$ mapping each pair of indexes (i,j) into another pair $(i',j') = \rho(i,j)$ such that $m^{(0)}_{i,j} = m^{(1)}_{\rho(i,j)} = m^{(1)}_{i',j'}$. (Since such permutation may not be unique, we let $\texttt{Perm}(M^{(0)}, M^{(1)})$ denote the smallest one under some standard encoding.)

Let A be an adversary with non-negligible advantage $\mathbf{Adv}^{\textbf{I-N}-anon}_{\pi,A}(k) = \epsilon(k)$. It suffices to show that, either A does not output an indistinguishable sampling pair, or there exist an adversary A^* with non-negligible advantage $\mathbf{Adv}^{\mathbf{N}-anon}_{\pi,A^*}(k)$ that breaks the $\mathbf{N}$-anonymity of π. First, assume we have such A which outputs a sampling pair $\langle G^{(0)}\rangle, \langle G^{(1)}\rangle$. We now show how to build a distinguishing algorithm D for ensembles $\mathcal{X}_0 \stackrel{\text{def}}{=} \{G^{(0)}(k,a)\}_{k,a}$, and $\mathcal{X}_1 \stackrel{\text{def}}{=} \{G^{(1)}(k,a)\}_{k,a}$. Let $D_{i,j}(\cdot)$ be the following algorithm parameterized by $i,j \in [n]$.

Distinguisher $D_{i,j}(x)$
Let $B_{i,j}$ be the following adversary:
Adversary $B_{i,j}(k)$
"Run adversary A, which outputs $M^{(0)}, M^{(1)}, \langle G^{(0)}\rangle, \langle G^{(1)}\rangle$.
Then, define algorithm $H_{i,j}(k,\cdot)$ as follows.
For each $u,v \in [n]$ define $H_{i,j}(k,\cdot)$ as

$$\langle H_{i,j}(k, m^{(1)}_{u,v})\rangle \stackrel{\text{def}}{=} \begin{cases} \langle G^{(1)}(k, m^{(1)}_{u,v})\rangle & \text{for } (u-1)n+v-1 < (i-1)n+j-1 \\ \text{“}\mathbf{Output}\ x\text{”} & \text{for } (u-1)n+v-1 = (i-1)n+j-1 \\ \langle G^{(0)}(k, m^{(1)}_{u,v})\rangle & \text{otherwise} \end{cases}$$

Output $M^{(0)}, M^{(1)}, \langle G^{(0)}\rangle, \langle H_{i,j}\rangle$.
From then on, give any input to A, and output what A outputs."
return $\mathbf{Exp}^{\textbf{I-N}-anon}_{\pi,B_{i,j}}(k)$

We claim that there exists $i^*, j^* \in [n]$, and $a^* \in V$ such that $D_{i,j}$ distinguishes ensembles $\mathcal{X}_0$ and $\mathcal{X}_1$. Wlog. fix the matrices $M^{(0)}, M^{(1)}$ output by A, which we assume belong to relation $R_{\mathbf{N}}$, and thus permutation $\rho = \texttt{Perm}(M^{(0)}, M^{(1)})$ is well defined. Clearly, for all i,j, $\Pr\left[\, D_{i,j}(G^{(0)}(k, m^{(1)}_{i,j})) = 1 \,\right] = \Pr\left[\, D_{i',j'}(G^{(1)}(k, m^{(1)}_{i',j'})) = 1 \,\right]$ if $(i'-1)n + j' = (i-1)n + j - 1$. Thus,

$$\begin{aligned} \epsilon(k) \;&=\; \mathbf{Adv}^{\textbf{I-N}-anon}_{\pi,A}(k) \;=\; 2\cdot \sum_{i,j\in[n]} \Big(\Pr\left[\, D_{i,j}(G^{(1)}(k, m^{(1)}_{i,j})) = 1 \,\right] - \Pr\left[\, D_{i,j}(G^{(0)}(k, m^{(1)}_{i,j})) = 1 \,\right]\Big) \\ &\quad + 2\cdot \Pr\left[\, D_{1,1}(G^{(0)}(k, m^{(0)}_{\rho^{-1}(1,1)}) = 1 \,\right] - 1 \\ &\le\; 2\cdot \sum_{i,j\in[n]} \left|\Pr\left[\, D_{i,j}(G^{(1)}(k, m^{(1)}_{i,j})) = 1 \,\right] - \Pr\left[\, D_{i,j}(G^{(0)}(k, m^{(1)}_{i,j})) = 1 \,\right]\right| + \mathbf{Adv}^{\textbf{I-N}-anon}_{\pi,B_{1,1}}(k) \end{aligned}$$

where we used that $m^{(1)}_{i,j} = m^{(0)}_{\rho(i,j)}$. Notice that $B_{1,1}$ is the adversary that truthfully simulates A, except when A outputs a sampling pair $\langle G^{(0)}\rangle, \langle G^{(1)}\rangle$, in which case $B_{1,1}$ outputs $\langle G^{(0)}\rangle, \langle G^{(0)}\rangle$ instead. We claim that for any such adversary $B_{1,1}$ there exist an adversary A^* (operating in the original experiment) with the same advantage, that is, $\mathbf{Adv}^{\mathbf{N}-anon}_{\pi,A^*}(k) = \mathbf{Adv}^{\textbf{I-N}-anon}_{\pi,B_{1,1}}(k)$. Before proving this claim, we show how to obtain the proposition using the claim. Let $(i^*, j^*) \in [n]^2$ be the indices for which the value in absolute value inside the above sum is maximized, and let $a^* = m^{(1)}_{i^*,j^*}$. Then,

$$\epsilon(k) \;\le\; 2n^2 \cdot \left|\Pr\left[\, D_{i^*,j^*}(G^{(1)}(k,a^*)) = 1 \,\right] - \Pr\left[\, D_{i^*,j^*}(G^{(0)}(k,a^*)) = 1 \,\right]\right| + \mathbf{Adv}^{\mathbf{N}-anon}_{\pi,A^*}(k)$$

Therefore, if $\epsilon(k)$ is non-negligible, then either there exist a distinguishing algorithm $D = D_{i^*,j^*}$ for $\mathcal{X}_0$ and $\mathcal{X}_1$ that succeeds with non-negligible probability on index a^*, or adversary A^* breaks the $\mathbf{N}$-anonymity of protocol π.

We now prove the claim that such A^* exists. Given $B_{1,1}$, we build adversary A^* as follows. Adversary A^* simulates $B_{1,1}$ until the latter outputs $M'^{(0)}, M'^{(1)}, \langle G'^{(0)}\rangle, \langle G'^{(1)}\rangle$. Assume wlog. that $M'^{(0)}, M'^{(1)}$ belong to $R_{\mathbf{N}}$ (otherwise abort) and thus $\rho = \mathtt{Perm}(M'^{(0)}, M'^{(1)})$ is well-defined. Then, A^* computes $\bar{m}^{*(0)}_{i,j} \stackrel{R}{\leftarrow} G^{(0)}(k, m'^{(0)}_{i,j})$ and $\bar{m}^{*(1)}_{\rho(i,j)} \leftarrow \bar{m}^{*(0)}_{i,j}$, for all $i,j \in [n]$, and then outputs the matrices $\bar{M}^{*(0)} = \{\bar{m}^{*(0)}_{i,j}\}_{i,j\in[n]}$ and $\bar{M}^{*(1)} = \{\bar{m}^{*(1)}_{i,j}\}_{i,j\in[n]}$. From then on, A^* simulates $B_{1,1}$ for the rest of the experiment. For the analysis, first notice that A^* ouput valid matrices $(\bar{M}^{*(0)}, \bar{M}^{*(1)}) \in R_{\mathbf{N}}$ since the pair $(M'^{(0)}, M'^{(1)})$ also belongs to $R_{\mathbf{N}}$. It remains to argue that the success probability of $B_{1,1}$, which runs in $E \stackrel{\text{def}}{=} \mathbf{Exp}^{\mathbf{I}\text{-}\mathbf{N}-anon}_{\pi,B_{1,1}}$, is as good as that of A^* in $E^* \stackrel{\text{def}}{=} \mathbf{Exp}^{\mathbf{N}-anon}_{\pi,A^*}$. This follows from observing that A^* perfectly simulates $B_{1,1}$ for experiment E^*, so adversary $B_{1,1}$ cannot distinguish whether is executed as part of A^* or inside E. In fact, since $\langle G'^{(0)}\rangle = \langle G'^{(1)}\rangle$, from the point of view of $B_{1,1}$ the distribution of matrix $\bar{M}^{(b)}$ (for any bit b) is identical in both experiments. Since $B_{1,1}$'s view depends solely on $\bar{M}^{(b)}$, the success probability of $B_{1,1}$ and A^* are thus the same. This concludes the proof of the proposition. ▌

Given any $\mathbf{I}$-$\mathbf{N}$-anonymous protocol π for $\mathbf{N} \in \{\mathsf{SUL}, \mathsf{RUL}, \mathsf{SA}, \mathsf{RA}\}$, the simple transformation consisting of encrypting (under a key-private encryption scheme [4]) each message under the public key of the recipient produces a protocol that can achieve a stronger anonymity notion. Indeed, next proposition simply shows that breaking the stronger notion gives raise to a *legal adversary* for the weaker notion $\mathbf{I}$-$\mathbf{N}$.

Proposition 4.5 Assume a semantically secure public-key encryption scheme exists [30]. Then I-SUL $\rightarrow$ UL, and I-SA $\rightarrow$ SA*. Moreover, if the encryption scheme is key-private [4], then I-RUL $\rightarrow$ UL, and I-RA $\rightarrow$ RA*. ▌

Proof of Proposition 4.5: We exhibit a simple black-box transformation $\theta^{(\cdot)}$ that, when applied to any $\mathbf{I}$-$\mathbf{N}$-anonymous protocol π, where $\mathbf{N}$ is either SUL,RUL,SA, or RA, produces a $\mathbf{N}'$-anonymous protocol θ^{π}, where $\mathbf{N}'$ is either UL,UL,SA*, or RA* respectively. This will prove the desired implications. The construction $\theta^{(\cdot)}$ is simple: given an input set of messages to send, each party encrypts (under the appropriate encryption scheme) each message under the intended recipient's public key, and use those as inputs to π; the local output is then the decryptions of the values received from π. To achieve security, the construction assumes the so-called public key infrastructure (as described in Appendix A) in which parties have access to authenticated copies of the public keys for all other parties. Formally, let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a semantic secure encryption scheme [30] (which in particular implies $\mathcal{E}$ is randomized) and IK-CPA [4], and let (pk_i, sk_i) denote the public/private key pair corresponding to party P_i. For any public key pk and message m we denote by $\mathcal{E}(pk, m; r)$ the encryption of m under public key pk using random string r.

We now describe protocol θ^{π} given any message-transmission protocol π. Each party P_i initially holds input $\{m_{i,j}\}_{j\in[n]}$.

1. For each message $m_{i,j}$, each party P_i computes the encryption $y_{i,j} \stackrel{R}{\leftarrow} \mathcal{E}(pk_j, m_{i,j})$ of $m_{i,j}$ under party P_j's public key.

2. Each party P_i, calls protocol π on input $\{y_{i,j}\}_{j\in[n]}$. Let $\{z_{\ell,i}\}_{\ell}$ be the lexicographically-sorted set that represents the party's local output returned by π.

3. Each party P_i computes the decryption $m'_\ell \stackrel{R}{\leftarrow} \mathcal{D}(sk_i, z_{\ell,i})$ of $z_{\ell,i}$ under using its private key, for all received messages $z_{\ell,i}$.

4. Each party P_i outputs $\{m'_\ell\}_\ell$ as the local output.

The implications stated in the claim are proven next. In what follows, we denote matrices with uppercase letters (say X), and their (i,j)-th elements by lowercase letters (say $x_{i,j}$).

I-RUL $\rightarrow$ UL: It suffices to show that given protocol $\tau \stackrel{\text{def}}{=} \theta^\pi$ and an arbitrary adversary A_τ attacking the UL-anonymity of τ, there exists an adversary A_π attacking the I-RUL-anonymity of π. The idea is to let A_π simulate the encryption and decryption phases of protocol τ for A_τ as follows. Adversary A_π on input k, it first executes $A_\tau(k)$. By assumption, $A_\tau(k)$ outputs a pair $(M^{(0)}, M^{(1)}) \in R_{\mathsf{UL}}$. Adversary A_π then generates a random key pair $(pk, sk) \stackrel{R}{\leftarrow} \mathcal{K}(k)$ and, for $d = 0, 1$, it computes $\langle \hat{G}^{(d)}(k, a)\rangle \stackrel{\text{def}}{=} \langle \mathcal{E}(pk, a; \cdot)\rangle$, where $\langle \mathcal{E}(pk, a; \cdot)\rangle$ denotes the *description* of the probabilistic algorithm that, when called on input a, outputs an encryption of a under pk.[8] Adversary A_π then computes new "left-or-right" matrices $\hat{M}^{(0)}, \hat{M}^{(1)}$ as follows: first, it select a random value $z \in V$; then A_π computes $\hat{m}^{(d)}_{i,j} \stackrel{R}{\leftarrow} z$ if $m^{(d)}_{i,j} \neq \emptyset$ and $\hat{m}^{(d)}_{i,j} \stackrel{R}{\leftarrow} \emptyset$ otherwise, for all $i, j \in [n]$ and $d = 0, 1$. The tuple $(\hat{M}^{(0)}, \hat{M}^{(1)}, \langle \hat{G}^{(0)}\rangle, \langle \hat{G}^{(1)}\rangle)$ is then output by A_π. From then on, A_π transparently follows A_τ's instructions while attacking π: it forwards all information received from the execution of π to adversary A_τ until A_τ outputs a bit b and stops, in which case A_π outputs the same and stops.[9]

We claim that, unless $\mathcal{AE}$ is not a IND-CPA or IK-CPA secure encryption scheme, A_π correctly simulates the experiment for A_τ. First, notice that the "left-or-right" matrix pair $\hat{M}^{(0)}, \hat{M}^{(1)}$ output by A_π belongs to R_{RUL} as long as the pair $(M^{(0)}, M^{(1)})$ output by A_τ belongs to R_{UL}. Now we show that the distribution obtained by the sampling from $\hat{G}^{(0)}, \hat{G}^{(1)}$ during the simulation of A_τ and the distribution of the inputs feed to subprotocol π while running a real execution of τ are computationally close. To see this, let $X = \bar{M}^{(b)}$ be the message matrix used as input to protocol π in $\mathbf{Exp}^{\mathsf{I-RUL}-anon}_{\pi,A_\pi}$; and let Y be the message matrix used as input to subprotocol π while executing $\tau = \theta^\pi$ in $\mathbf{Exp}^{\mathsf{UL}-anon}_{\tau,A_\tau}$. Clearly, by definition of the experiments, $x_{i,j} = \mathcal{E}(pk, \hat{m}^{(b)}_{i,j}) = \mathcal{E}(pk, z)$ if $m^{(b)}_{i,j}$ is not empty ($x_{i,j} = \emptyset$ otherwise) for some public key pk and value z chosen anew by A_π, and $y_{i,j} = \mathcal{E}(pk_j, m^{(b)}_{i,j})$ if $m^{(b)}_{i,j}$ is not empty ($y_{i,j} = \emptyset$ otherwise) where pk_j is the public key for party P_j. By a standard hybrid argument, any advantage $\epsilon(k)$ in distinguishing inputs X from Y by A_τ can be transformed into an advantage of at least $\epsilon(k)/(2n^2)$ in breaking the IND-CPA security of the encryption scheme $\mathcal{AE}$, or an advantage of at least $\epsilon(k)/(2n^2)$ in breaking the IK-CPA security of the same scheme. A similar argument shows that A_π outputs a legal sampling pair $\hat{G}^{(0)}(k, a) = \hat{G}^{(1)}(k, a) = \langle \mathcal{E}(pk, a; \cdot)\rangle$ if $(pk, sk) \stackrel{R}{\leftarrow} K(k)$. The proof for the case I-RA $\rightarrow$ RA* is essentially the same.

For the cases I-SUL $\rightarrow$ UL and I-SA $\rightarrow$ SA* the proof can be done in similar way as above. In these cases, however, it is possible to prove the correct simulation of A_τ from *only* the IND-CPA security of the encryption scheme (no IK-CPA security is needed). To illustrate this, we outline the proof of I-SUL $\rightarrow$ UL. (The same proof works for I-SA $\rightarrow$ SA*.) The simulation is analogous to the one above with the following exceptions: adversary A_π chooses "left-or-right" matrices $\hat{M}^{(0)}, \hat{M}^{(1)}$ by first selecting $z \stackrel{R}{\leftarrow} V$, and then computing, for all $i, j \in [n]$ and $d = 0, 1$, $\hat{m}^{(d)}_{i,j} \leftarrow \langle j, z\rangle$ if $m^{(d)}_{i,j}$ is non-empty and $\hat{m}^{(d)}_{i,j} \stackrel{R}{\leftarrow} \emptyset$ otherwise. Clearly, if $(M^{(0)}, M^{(1)}) \in R_{\mathsf{UL}}$, then $(\hat{M}^{(0)}, \hat{M}^{(1)}) \in R_{\mathsf{SUL}}$. To achieve a correct simulation, A_π sets the

[8] In the description of the algorithm $\hat{G}^{(d)}$, a denotes a *variable* which is instantiated when the algorithm is evaluated.

[9] Since we do not allow A_τ to corrupt receivers, there is no need to simulate the decryption of the values received by the parties from π. If needed, it would be straightforward though.

sampling pair $\langle \hat{G}^{(0)} \rangle, \langle \hat{G}^{(1)} \rangle)$ to $\langle \hat{G}^{(d)}(k, \langle t, a \rangle) \rangle \stackrel{\text{def}}{=} \langle \mathcal{E}(PK[t], a; \cdot) \rangle$, for $d = 0, 1$, where PK is a table whose index t contains the public key for party P_t. (Notice that each sampling algorithm must include the table PK in its description). For the analysis, correct simulation of A_τ by A_π can be easily argued from the IND-CPA security. Indeed, for each column $j \in [n]$, applying sampling algorithm $\hat{G}^{(d)}$ on the (i,j)-th element of $\hat{M}^{(d)}$ generates $\hat{G}^{(d)}(k, \langle j, m_{i,j}^{(d)} \rangle) = \mathcal{E}(PK[j], m_{i,j}^{(d)}; \cdot) = \mathcal{E}(pk_j, m_{i,j}^{(d)})$ which follows the same distribution as the inputs of subprotocol π in an actual execution of τ. Proving that the sampling pair output by A_π is legal is also simpler. Each "left-or-right" matrix $\hat{M}^{(d)}$, for $d = 0, 1$, contains no duplicate elements per row, therefore each sampling algorithm is guaranteed to be evaluated over different values per row. Therefore, no indistinguishability condition for the sampling algorithms is needed among those indexes – indistinguishability must only hold when evaluated in elements of the same matrix column, say j. In that case, however, the definition of $\hat{G}^{(d)}$, for $d = 0, 1$, guarantees that the same public key pk_j is used, and IND-CPA security suffices to prove the algorithms $(\hat{G}^{(0)}, \hat{G}^{(1)})$ legal. This concludes the proof of the claim. ▮

Proof of Lemma 4.2: It follows directly from combining Proposition 4.4 and 4.5. ▮

4.2 Implications that Require "Dummy Traffic"

In this section, we show that notions UL, SA*, RA*, SRA, and UO are equivalent under reductions that involve sending dummy traffic. Notions SUL and SA, as well as RUL and RA are also equivalent.

Let `D2Sink` be the following protocol transformation. Given a message-transmission protocol π, output another protocol that operates like π but where each sender transmits additional empty messages *to a fixed party* (the "sink") until the sender's total number equals a given constant $\mu_{\mathbf{N}}$. The next proposition shows `D2Sink` can be used to achieve stronger notions of anonymity.

Proposition 4.6 Assume the total number of messages in any protocol for the notions $\mathsf{SA}, \mathsf{SA}^*$, and UO is upper bounded by a publicly known value $\mu_{\mathbf{N}}$. Then, $\mathsf{SUL} \rightarrow \mathsf{SA}$, $\mathsf{UL} \rightarrow \mathsf{SA}^*$, and $\mathsf{RA}^* \rightarrow \mathsf{UO}$. ▮

Proof of Proposition 4.6: The three implications are proven using the same black-box transformation `D2Sink` which maps n-party PPT protocols into other n-party protocols.[10] If applied to any $\mathbf{N}$-anonymous protocol, this transformation (where $\mathbf{N}$ is either SUL,UL, or RA*) outputs a $\mathbf{N}'$-anonymous protocol (where $\mathbf{N}'$ is either SA,SA*, or UO respectively). Informally speaking, the construction underlying `D2Sink` relies on "dummy messages". Given as input an arbitrary message-transmission protocol π, `D2Sink` outputs a protocol $\delta^{\pi}_{\mathtt{D2Sink}}$ that essentially operates like π but inputs are "padded" with appropriately-addressed null-valued messages. Indeed, in $\delta^{\pi}_{\mathtt{D2Sink}}$, each party's input (which is a set of messages to send) is appended with a certain number of *null-valued messages* whose destination is party P_s, called the "sink", whose identity is fixed for all parties. (Alternatively, P_s can be represented by some non-existent party – the same for all senders – whose traffic gets discarded.) Then protocol π is invoked on the extended inputs which are delivered as expected. Party P_s then discards all null-valued messages it receives. We stress that, in this construction, how to use the "dummy messages" does not depend on the protocol π input to `D2Sink`. The construction does assume, however, that for each notion $\mathbf{N} \in \{\mathsf{SA}, \mathsf{SA}^*, \mathsf{UO}\}$ there exists a quantity $\mu_{\mathbf{N}}$ that bounds the total number of messages that can be sent by any protocol achieving the notion. For concreteness' sake, protocol $\delta^{\pi}_{\mathtt{D2Sink}}$ is show next. Here, each party P_i initially holds input vector $(m_{i,j})_{j \in [n]}$.

1. Each party P_i, computes the number of "dummy messages" $\ell_i \leftarrow \mu_{\mathbf{N}} - \sum_{j \in [n]} |m_{i,j}|$ needed.

[10] `D2Sink` stands for sending "dummy messages to (single) sink".

2. Each party P_i, sets $x_{i,s} \leftarrow m_{i,s} \uplus (\uplus_{i=1\dots\ell_i}\{\perp\})$, and $x_{i,j} \leftarrow m_{i,j}$ if $j \neq s$.

3. Each party P_i, calls protocol π on input $(x_{i,j})_{j\in[n]}$. Let $\{z_{\ell,i}\}_\ell$ be the lexicographically-sorted multiset that represents local output returned by π to P_i.

4. If $i = s$, party P_i discard any element $z_{\ell,s} = \perp$, and locally output the remaining elements. Otherwise, party P_i outputs $\{z_{\ell,i}\}_\ell$ as the local output.

We now prove $\mathsf{SUL} \rightarrow \mathsf{SA}$. It suffices to show that given protocol $\nu \stackrel{\text{def}}{=} \delta^{\pi}_{\mathtt{D2Sink}}$ and an arbitrary adversary A_ν attacking the SA-anonymity of ν, there exists an adversary A_π attacking the SUL-anonymity of π. The idea is to let A_π simulate the operation of protocol ν for A_ν as follows. Adversary A_π on input k, it first executes $A_\nu(k)$. By assumption, $A_\nu(k)$ outputs a pair $(M^{(0)}, M^{(1)}) \in R_{\mathsf{SA}}$. Adversary A_π then generates the appropriate dummy messages for each party P_i by essentially emulating the operation of ν. Namely, for $d = 0, 1$, A_π creates vector $(\hat{m}^{(d)}_{i,j})_j$ from each party P_i's input $(m^{(d)}_{i,j})_{j\in[n]}$ by following steps 1-2 of protocol ν. The pair $(\hat{M}^{(0)}, \hat{M}^{(1)})$ is then output by A_π. From then on, A_π transparently follows A_ν's instructions: it forwards all information received from the execution of π to adversary A_ν and viceversa, until A_ν outputs a bit b and stops, in which case A_π outputs the same and stops. Correct simulation follows from observing that the total number of "dummy messages" sent to P_s is the same no matter what bit b is set in $\mathbf{Exp}^{\mathsf{SUL}-anon}_{\pi,A_\pi}$. By construction, for $d = 0, 1$ the number of messages sent by P_i as instructed by $\hat{M}^{(d)}$ is $f_i = \sum_{j\in[n]} |\hat{m}^{(d)}_{i,j}| = \mu_{\mathbf{N}}$; the total number of messages is then $n\mu_{\mathbf{N}} = \sum_{i\in[n]} f_i = \sum_{i,j\in[n]} |m^{(d)}_{i,j}| + \sum_{i\in[n]} \ell^{(d)}_i$. But since $\sum_{i,j\in[n]} |m^{(0)}_{i,j}| = \sum_{i,j\in[n]} |m^{(1)}_{i,j}|$ then $\sum_{i\in[n]} \ell^{(0)}_i = \sum_{i\in[n]} \ell^{(1)}_i$. Moreover, since all dummy messages sent to P_s are equal to "$\perp$", $(\hat{M}^{(0)}, \hat{M}^{(1)}) \in R_{\mathsf{SUL}}$.

The proof for $\mathsf{UL} \rightarrow \mathsf{SA}^*$ is essentially identical to the one above. The proof for $\mathsf{RA}^* \rightarrow \mathsf{UO}$ is also very similar but slightly more general, as it holds even under adversaries that output message matrices for which $\sum_{i,j\in[n]} |m^{(0)}_{i,j}| \neq \sum_{i,j\in[n]} |m^{(1)}_{i,j}|$, as long as both quantities are upper bounded by a constant μ_{UO}. ▮

Similarly, let `D2All` be the transformation that instructs senders to transmit one dummy message to everyone else per each valid message to be sent. `D2All` is used to prove the following implications.

Proposition 4.7 $\mathsf{RUL} \rightarrow \mathsf{RA}$, $\mathsf{UL} \rightarrow \mathsf{RA}^*$, and $\mathsf{SA}^* \rightarrow \mathsf{SRA}$. ▮

Proof of Proposition 4.7: The proof follows the same structure as the one of Proposition 4.6. Given an arbitrary message-transmission protocol π, protocol $\delta^{\pi}_{\mathtt{D2All}}$ works as follows: for each message $m_{i,j}$ in P_i's input, P_i sends a single new null-valued message to all other P_k, $k \neq j$. Then protocol π is invoked on the modified inputs. From the output received by π, each party P_i then discards all received null-valued messages. Let `D2All` be the transform that maps a message-transmission protocol π to another message-transmission protocol $\delta^{\pi}_{\mathtt{D2All}}$. Protocol $\delta^{\pi}_{\mathtt{D2All}}$ is described next. As opposed to transformation `D2Sink`, this construction does not assume any bounds on the total number of messages exchanged by the parties. Each party P_i initially holds input vector $(m_{i,j})_{j\in[n]}$.

1. Each party P_i, computes the number of "dummy messages" $\ell_{i,j} \leftarrow \sum_{k\in[n]\setminus\{j\}} |m_{i,j}|$ needed to send to party P_j.

2. Each party P_i, sets $x_{i,j} \leftarrow m_{i,j} \uplus (\uplus_{i=1\dots\ell_{i,j}}\{\perp\})$.

3. Each party P_i, calls protocol π on input $(x_{i,j})_{j\in[n]}$. Let $\{z_{\ell,i}\}_\ell$ be the lexicographically-sorted multiset that represents the local output returned by π to P_i.

4. Each party P_i discard any element $z_{\ell,s} = \perp$, and locally output the remaining elements.

We now prove RUL $\to$ RA. Let π be a message-transmission protocol, and $\kappa \stackrel{\text{def}}{=} \delta^{\pi}_{\texttt{D2All}}$. We show that given an arbitrary adversary A_κ attacking the RA-anonymity of κ, there exists an adversary A_π attacking the RUL-anonymity of π. Adversary A_π simulates the operation of protocol κ for A_κ as follows. First, adversary A_π, on input k, obtains a pair $(M^{(0)}, M^{(1)})$ from running $A_\kappa(k)$. From it, A_π generates two new matrices $X^{(0)} = (x^{(0)}_{i,j})_{i,j\in[n]}$ and $X^{(1)} = (x^{(1)}_{i,j})_{i,j\in[n]}$, by adding the appropriate dummy messages for each party P_i according to steps 1-2 of protocol $\delta_{\texttt{D2All}}$ (as described above). Then A_π outputs $(X^{(0)}, X^{(1)})$ as the message matrix pair for experiment $\mathbf{Exp}^{\mathsf{RUL}-anon}_{\pi,A_\pi}$. From then on, A_π transparently follows A_κ's instructions: it forwards all information received from the execution of π to adversary A_κ and viceversa, until A_ν outputs a bit b and stops, in which case A_π outputs the same and stops.

We argue that A_π is a good adversary for $\mathbf{Exp}^{\mathsf{RUL}-anon}_{\pi,A_\pi}$ if A_κ is good for $\mathbf{Exp}^{\mathsf{RA}-anon}_{\kappa,A_\kappa}$. It suffices to show that $(X^{(0)}, X^{(1)}) \in R_{\mathsf{RUL}}$ if $(M^{(0)}, M^{(1)}) \in R_{\mathsf{RA}}$. At this point, we need to define some quantities. For $d = 0, 1$, we denote by $f^{(d)}_i = \sum_{j\in[n]} |m^{(d)}_{i,j}|$ (resp. $\hat{f}^{(d)}_i = \sum_{j\in[n]} |x^{(d)}_{i,j}|$) the total number of messages sent by P_i as encoded by $M^{(d)}$ (resp. $X^{(d)}$). Similarly, $\ell^{(d)}_{i,j}$ denotes the number of "dummy messages" send by P_i to P_j as encoded by $X^{(d)}$, and $\ell^{(d)}_i = \sum_{j\in[n]} \ell^{(d)}_{i,j}$ the total number of such messages. It is easy to see that $\hat{f}^{(d)}_i = \sum_{j\in[n]}(|m^{(d)}_{i,j}| + \ell^{(d)}_{i,j}) = f^{(d)}_i + \ell^{(d)}_i$, and $\hat{f}^{(d)}_i = n \cdot f^{(d)}_i$. So $\ell^{(d)}_i = (n-1) \cdot f^{(d)}_i$. Moreover, since $(M^{(0)}, M^{(1)}) \in R_{\mathsf{RA}}$ then, in particular $(M^{(0)}, M^{(1)}) \in R_{\mathsf{U}}$, which implies $f^{(0)}_i = f^{(1)}_i$, and $\ell^{(0)}_i = \ell^{(1)}_i$, for all $i \in [n]$. Combining these, the multiset of messages sent by P_i is then

$$\begin{aligned}
\uplus_{j\in[n]} x^{(0)}_{i,j} &= \uplus_{j\in[n]} \left(m^{(0)}_{i,j} \uplus (\uplus_{i=1\ldots\ell^{(0)}_{i,j}} \{\perp\}) \right) = \uplus_{j\in[n]} m^{(0)}_{i,j} \uplus (\uplus_{k=1\ldots\ell^{(0)}_i} \{\perp\}) \\
&= \uplus_{j\in[n]} m^{(1)}_{i,j} \uplus (\uplus_{k=1\ldots\ell^{(1)}_i} \{\perp\}) = \uplus_{j\in[n]} \left(m^{(1)}_{i,j} \uplus (\uplus_{i=1\ldots\ell^{(1)}_{i,j}} \{\perp\}) \right) = \uplus_{j\in[n]} x^{(1)}_{i,j}
\end{aligned}$$

and $(X^{(0)}, X^{(1)}) \in R_{\mathsf{U}}$ follows.

To argue that $(X^{(0)}, X^{(1)}) \in R^T_\Sigma$, it suffices to see that the total number of messages ("regular" and and "dummy" messages) to be received by any party P_j according to $X^{(d)}$, $d = 0, 1$, is $\sum_{i\in[n]} |x^{(d)}_{i,j}| = \sum_{i\in[n]}(|m^{(d)}_{i,j}| + \ell^{(d)}_{i,j}) = \sum_{i,j\in[n]} |m^{(d)}_{i,j}| = |M^{(d)}|$. But then, $|M^{(0)}| = |M^{(1)}|$ is implied by $(M^{(0)}, M^{(1)}) \in R_{\mathsf{U}}$, and the result follows.

The proof for UL $\to$ RA* is analogous (indeed, simpler since we need to prove the matrix pair output by the UL adversary satisfies R_Σ instead of R_{U}). A similar argument also proves SA* $\to$ SRA. We notice that, in this latter case, the proof relies on the condition $|M^{(0)}| = |M^{(1)}|$ guaranteed by any SRA-adversary. ∎

4.3 Message Overhead and Optimality of the Transformations

The black-box transformations `D2Sink` of Proposition 4.6 and `D2All` of Proposition 4.7 output protocols that use "dummy" messages (those whose value is "$\perp$" which are ultimately discarded). These messages increase the communication complexity of the protocol, so it is interesting to ask if there are better solutions, possibly based on cryptographic tools. Interestingly, we show that the single transformations `D2Sink` and `D2All` described in previous section cannot be substantially improved, even in the presence of PKI.

Thus, we explore the question of whether more *message efficient* transformations exist, in terms of generating protocols where fewer messages (dummy or not) are sent overall.[11] For simplicity, we consider transformations where the input protocol is invoked via a black-box call only once; the general case is discussed at the end of the section.

Let $\mathtt{T}$ be a transformation that maps a protocol ω into another protocol $\delta^{\omega}_{\mathtt{T}}$. We measure message overhead by counting the number of extra messages that any protocol $\delta^{\omega}_{\mathtt{T}} \stackrel{\text{def}}{=} \mathtt{T}(\omega)$ adds on the underlying (black-box) protocol π. Concretely, given two transformations $\mathtt{T}_1, \mathtt{T}_2$, we say $\mathtt{T}_1$ has less message overhead than $\mathtt{T}_2$ if protocols $\delta^{\omega}_{\mathtt{T}_1} = \mathtt{T}_1(\omega)$ and $\delta^{\omega}_{\mathtt{T}_2} = \mathtt{T}_2(\omega)$ when executed on the same input matrix M require subprotocol ω to send t_1 (resp. t_2) messages when invoked as part of $\delta^{\omega}_{\mathtt{T}_1}$ (resp. $\delta^{\omega}_{\mathtt{T}_2}$), where $t_1 < t_2$ for any protocol ω. More formally, let $M = (m_{i,j})_{i,j\in[n]}$ be a message matrix, and denote by $\delta^{[\cdot]}_{\mathtt{T}}(M) \in \mathcal{M}_{n\times n}(\mathcal{P}(V))$ the message matrix on which the black-box protocol (say ω) is invoked via a black-box call during the execution of $\delta^{\omega}_{\mathtt{T}}$ on input matrix M. We stress that once M is fixed, matrix $\delta^{[\cdot]}_{\mathtt{T}}(M)$ is well-defined, independently of the message-transmission protocol ω, as ω is invoked as black-box by $\delta^{\omega}_{\mathtt{T}}$ exactly once.

Definition 4.8 Let $(\mathbf{N}', \mathbf{N}) \in \{(\mathsf{SUL}, \mathsf{SA}), (\mathsf{RUL}, \mathsf{RA}), (\mathsf{UL}, \mathsf{SA}^*), (\mathsf{UL}, \mathsf{RA}^*), (\mathsf{RA}^*, \mathsf{SRA}), (\mathsf{SA}^*, \mathsf{SRA})\}$, and $\mathtt{T}$ be any transformation underlying implication $\mathbf{N}' \to \mathbf{N}$. The *message overhead* of $\mathtt{T}$ is $\mathtt{ovh}(\mathtt{T}) \stackrel{\text{def}}{=} \max_M \left\{ \left|\delta^{[\cdot]}_{\mathtt{T}}(M)\right| / |M| \right\}$ where the maximum is taken over all (allowed) non-empty message matrices M for notion $\mathbf{N}$. ▌

It is easy to see that, under the assumption that the total number of messages sent is at most $\mu_{\mathbf{N}}$, $\mathtt{ovh}(\mathtt{D2Sink}) = n \cdot \mu_{\mathbf{N}}$. Similarly, but under no assumptions, $\mathtt{ovh}(\mathtt{D2All}) = n$. The next two propositions show that we cannot do better. The proof is by contradiction which is derived from the fact that if there are "too few" messages sent by a party, the underlying black-box protocol may no longer be invoked in a secure way. For Proposition 4.10, the construction and analysis are similar but considering the number of messages *received* by any party.

Proposition 4.9 $\mathtt{D2Sink}$ is optimal for $\mathsf{SUL}{\to}\mathsf{SA}$, $\mathsf{UL}{\to}\mathsf{SA}^*$, and $\mathsf{RA}^*{\to}\mathsf{UO}$. ▌

Proposition 4.10 $\mathtt{D2All}$ is optimal for $\mathsf{RUL}{\to}\mathsf{RA}$, $\mathsf{UL}{\to}\mathsf{RA}^*$, and $\mathsf{SA}^*{\to}\mathsf{SRA}$. ▌

We now proceed to prove the above propositions.

Proof of Proposition 4.9: By contradiction. Assume there exists a transformation $\bar{\mathtt{T}}$ that proves the implication $\mathsf{SUL} \to \mathsf{SA}$ but for which $\mathtt{ovh}(\bar{\mathtt{T}}) < n\mu_{\mathsf{SA}}$. That is, on input any arbitrary SUL-anonymous protocol π, transformation $\bar{\mathtt{T}}$ outputs an SA-anonymous protocol $\bar{\mathtt{T}}(\pi) = \delta^{\pi}_{\bar{\mathtt{T}}}$. Now, let π be a SUL-anonymous protocol and π' be identical to π with the exception that each party P_i also broadcasts the message "sending f_i messages", where f_i is the number of messages that P_i has been instructed to send, that is, $f_i = |\uplus_{j\in[n]} m^{(b)}_{i,j}|$ (where $M^{(b)} = (m_{i,j})_{i,j\in[n]}$ is the corresponding message matrix). Notice that such π' is SUL-anonymous. We then consider the adversary A^*, attacking the SA-anonymity of $\delta^{\pi'}_{\bar{\mathtt{T}}}$, that works as follows. On input $k \in \mathbb{N}$, it outputs two matrices: (a) $M^{(0)}$, which is chosen at uniformly at random among all message matrices with exactly $\sum_{i,j} |m^{(0)}_{i,j}| = \mu_{\mathsf{SA}}$ messages to send. (b) $M^{(1)}$, which contains a single randomly selected row i^* for which $m^{(1)}_{i^*,j} = \uplus_{i\in[n],j} m^{(0)}_{i,j}$, and for all rows $i \neq i^*$, $m^{(1)}_{i,j} = \emptyset$ (that is, in world 1 party P_{i^*} is the

[11] Recall that we say a message m is *sent* by a message-transmission protocol Π if m is an element of the message matrix given to the protocol Π as input. This message should not be confused with the *packets* sent over the point-to-point communication channels between the parties as the result of a particular implementation of Π.

only sender but it sends to P_j the same set of messages P_j would receive if it were in world 0). Then, A^* waits for the message "sending f messages" from P_{i^*}: if $f < \mu_{\mathsf{SA}}$ outputs 0, otherwise outputs 1. Then A^* halts.

We argue that A^* is breaks the SA-anonymity of $\delta_{\bar{\mathtt{T}}}^{\pi'}$ with non-negligible probability. Clearly, $(M^{(0)}, M^{(1)}) \in R_{\mathsf{SA}}$. Adversary A^* will distinguish the execution of $\delta_{\bar{\mathtt{T}}}^{\pi'}$ on input $M^{(0)}$ from the one on input $M^{(1)}$ by examining the execution of subprotocol π' on those inputs. To see this, let $X^{(d)} \stackrel{\text{def}}{=} \delta_{\bar{\mathtt{T}}}^{[\cdot]}(M^{(d)})$, for $d = 0, 1$, denote the input matrix for subprotocol π' when $\delta_{\bar{\mathtt{T}}}^{\pi'}$ runs on input $M^{(d)}$. (As usual, we use $x_{i,j}^{(d)}$ to denote the (i,j)-th element of $X^{(d)}$). Assume (for now) that $\left|\delta_{\bar{\mathtt{T}}}^{[\cdot]}(M^{(d)})\right|$ is constant when seen as function of $|M^{(d)}|$. If $\mathtt{ovh}(\bar{\mathtt{T}}) < n\mu_{\mathsf{SA}}$ then it must be the case that $|X^{(d)}| < n\mu_{\mathsf{SA}}$. This, in turn, implies there must exist a sender $P_{i'}$ that sends $\sum_j |x_{i',j}^{(0)}| < \mu_{\mathbf{N}}$ messages using π'. On the other hand, also by assumption, during the execution of $\bar{\mathtt{T}}(\pi') = \delta_{\bar{\mathtt{T}}}^{\pi'}$, all communication is done via π', ie. $\delta_{\bar{\mathtt{T}}}$ is non-interactive. In consequence, P_{i^*}'s input to subprotocol π' is computed by $\delta_{\bar{\mathtt{T}}}$ solely on P_{i^*}'s current input $(m_{i^*,j}^{(b)})_{j\in[n]}$ and random coins, and any publicly known information. It follows that, $|\biguplus_{j\in[n]} x_{i^*,j}^{(1)}| \geq |\biguplus_{i,j\in[n]} m_{i,j}^{(0)}| = \mu_{\mathsf{SA}}$ (ie. P_{i^*} must send at least μ_{SA} messages via π') otherwise protocol $\delta_{\bar{\mathtt{T}}}^{\pi'}$ is not a correct message-transmission protocol. Thus, with probability at least $1/n$ (over the choice of i^*), $i^* = i'$, and A^* successfully distinguishes the two executions.

We conclude showing that $\left|\delta_{\bar{\mathtt{T}}}^{[\cdot]}(M)\right|$ is a constant function of $|M|$ if $\bar{\mathtt{T}}$ is a transformation from SUL to SA. The proof is by contradiction. Assume there exists matrices M' and M'' such that $|M'| < |M''| \leq \mu_{\mathsf{SA}}$ but $\left|\delta_{\bar{\mathtt{T}}}^{[\cdot]}(M')\right| < \left|\delta_{\bar{\mathtt{T}}}^{[\cdot]}(M'')\right|$. From the definition of black-box protocol, we know that in protocol $\delta_{\bar{\mathtt{T}}}^{\pi'}$, each sender P_i on input vector $m_{i,*} = (m_{i,j})_{j\in[n]}$ computes a new vector of messages $x_{i,*} = (x_{i,j})_{j\in[n]}$ which is then used as i-th input when calling subprotocol π'. Let us denote this computation by $x_{i,*} = \delta_{\bar{\mathtt{T}}}(m_{i,*})_i$. Thus, in particular, $x'_{i,*} = \delta_{\bar{\mathtt{T}}}(m'_{i,*})_i$, and $x''_{i,*}$ for input $m''_{i,*}$. Since protocol π' is only SUL-anonymous, it must be that $\sum_{j\in[n]} |x'_{i,j}| = \sum_{j\in[n]} |x''_{i,j}|$, for any two inputs $m'_{i,*}$ and $m''_{i,*}$, otherwise protocol $\delta_{\bar{\mathtt{T}}}^{\pi'}$ cannot longer be assumed secure. Moreover, since $\delta_{\bar{\mathtt{T}}}$ is non-interactive, such value $\sum_{j\in[n]} |x'_{i,j}|$ must be constant, say $c_i > 0$. This implies that $\left|\delta_{\bar{\mathtt{T}}}^{[\cdot]}(M')\right| = \sum_{i\in[n]} c_i = \left|\delta_{\bar{\mathtt{T}}}^{[\cdot]}(M'')\right|$ which contradicts the hypothesis. Thus, there exists a constant $c = \sum_{i\in[c]} c_i$, such that $\left|\delta_{\bar{\mathtt{T}}}^{[\cdot]}(M)\right| = c$.

Similar arguments prove the optimality of the transform for the implications UL $\rightarrow$ SA* and RA* $\rightarrow$ UO. ▮

Proof of Proposition 4.10: We focus on the case RUL$\rightarrow$RA first. The proof is by contradiction. As before, we assume there exists a transformation $\bar{\mathtt{T}}$ that proves the implication RUL $\rightarrow$ RA for which $\mathtt{ovh}(\bar{\mathtt{T}}) < n$. That is, on input any arbitrary RUL-anonymous protocol π, transformation $\bar{\mathtt{T}}$ outputs an RA-anonymous protocol $\bar{\mathtt{T}}(\pi) = \delta_{\bar{\mathtt{T}}}^{\pi}$. Now, let π be a RUL-anonymous protocol and π' be identical to π with the exception that each party P_i also broadcasts the message "received g_i messages", where g_i is the number of messages that P_i has received after π has ended, that is, $g_j = |\biguplus_{i\in[n]} m_{i,j}^{(b)}|$ (where $M^{(b)} = (m_{i,j})_{i,j\in[n]}$ is the corresponding message matrix). Notice that such π' is RUL-anonymous. Now, if D2All is not optimal, there exists a transformation $\bar{\mathtt{T}}$ from RUL to RA with $\mathtt{ovh}(\bar{\mathtt{T}}) = \max_M \left\{\left|\delta_{\bar{\mathtt{T}}}^{[\cdot]}(M)\right| / |M|\right\} < n$. Let M^* the matrix on which the maximum is reached. Then, $\left|\delta_{\bar{\mathtt{T}}}^{[\cdot]}(M^*)\right| < n \cdot |M^*|$. Notice this implies that there exists a party $P_{i'}$ that receives $\sum_j |x_{i',j}^{(0)}| < |M^*|$ messages using π'.

We then consider an adversary A^* which attacks the RA-anonymity of $\delta_{\bar{\mathsf{T}}}^{\pi'}$. A^* works as follows. On input $k \in \mathbb{N}$, it outputs $(M^{(0)}, M^{(1)})$ that satisfy (a) $M^{(0)} = M^*$, and (b) $M^{(1)}$ contains a single uniformly selected at random column j^* for which $m_{i,j^*}^{(1)} = \uplus_{j\in[n],i} m_{i,j}^{(0)}$, and for all other columns $j \neq j^*$, $m_{i,j}^{(1)} = \emptyset$ (that is, party P_{j^*} receives from P_i all messages sent by P_i in world 0, even those addressed to other recipients). Then, A^* waits for the message "received g messages" from P_{j^*}: if $g < |M^*|$ outputs 0, otherwise outputs 1. Then A^* halts.

We argue that A^* is breaks the RA-anonymity of $\delta_{\bar{\mathsf{T}}}^{\pi'}$ with non-negligible probability. Clearly, by construction, $(M^{(0)}, M^{(1)}) \in R_{\mathsf{RA}}$. We now argue that A^* can distinguish the execution of $\delta_{\bar{\mathsf{T}}}^{\pi'}$ on input $M^{(0)}$ from the one on input $M^{(1)}$ by examining the execution of subprotocol π' on those inputs. For $d = 0, 1$, let $X^{(d)} \stackrel{\text{def}}{=} \delta_{\bar{\mathsf{T}}}^{[\cdot]}(M^{(d)})$ denote the input matrix for subprotocol π' when $\delta_{\bar{\mathsf{T}}}^{\pi'}$ runs on input $M^{(d)}$. (As usual, we use $x_{i,j}^{(d)}$ to denote the (i,j)-th element of $X^{(d)}$). Recall that, $\left|\delta_{\bar{\mathsf{T}}}^{[\cdot]}(M^*)\right| < n \cdot |M^*|$. In our attack, this implies that, in world 0 there exists a party $P_{i'}$ that receives $\sum_j |x_{i',j}^{(0)}| < |M^*|$ messages using π'. On the other hand, by the non-inactivity of $\delta_{\bar{\mathsf{T}}}$ *after* the call to π', the correctness of $\delta_{\bar{\mathsf{T}}}^{\pi'}$, and since π' is called only once, it follows that, $X^{(1)}$ must satisfy $| \uplus_{i\in[n]} x_{i,j^*}^{(1)}| \geq | \uplus_{i,j\in[n]} m_{i,j}^{(0)}| = |M^*|$, ie. P_{j^*} must receive at least $|M^*|$ distinct messages in π'. Thus, with probability at least $1/n$ (over the choice of j^*), $j^* = j'$, and A^* successfully distinguishes the two executions.

Similar arguments prove the optimality of the transform for the implications $\mathsf{UL} \to \mathsf{RA}^*$ and $\mathsf{SA}^* \to \mathsf{SRA}$. ∎

UPPER BOUND PER SENDER: A similar analysis holds if a bound $\hat{\mu}_{\mathbf{N}}$ on the number of messages *per sender* is assumed instead, for SA and SA*-anonymity. (We stress that the implication $\mathsf{SA} \to \mathsf{SA}^*$ of Lemma 4.2 is preserved under this restriction). In this case the overhead is $n \cdot \hat{\mu}_{\mathbf{N}}$, which is also optimal. This formulation, although more restrictive, can be more suitable for certain applications.[12] From a theoretical point, however, it is not clear if there is any advantage to this formulation over the one presented above.

SINGLE VS. MULTIPLE BLACK-BOX CALLS: If we consider transformations that output protocols that invoke the input (black-box) protocol more than once, then is it possible to prove that the optimal overhead is n. A protocol δ^π that achieves this is the one that uses a *secure multiparty computation protocol* (eg. [6]) to compute $|M|$ using π as communication channel; then, each party calls ensures it sends $|M|$ messages via π by adding sufficient dummy messages. Even though such a secure multiparty protocol can be computed with constant number of invocations to π [2] (and thus, $\mathcal{O}(n^2)$ messages), it is likely that invoking π more than once will render the resulting protocol impractical.

5 On the Anonymity of Previous Protocols

The ultimate purpose of a definition is to be used to properly characterize the security of concrete protocols. Accordingly, we revisit the security of known constructions based on broadcast channels [8], DC-nets or anonymous networks [15, 32, 54], and mix-nets [33, 44, 24]. In Section 5, we examine the basic construction of Blaze et al. [8], which is based on broadcast channels, and we argue it can be shown *strong receiver anonymous*. We also discuss the DC-nets of [32] and sketch how the construction there can be proven *sender anonymous*. Finally, we highlight sufficient conditions to prove the *strong receiver anonymity* of mix-net constructions based on shuffles [33, 44]. By combining the constructions that underlie the implications

[12] Upper bounds on the number of messages sent *per party* may help to prevent certain *flooding* attacks against mix nets [34, 52].

of previous sections, we obtain anonymous protocols provably secure under the strongest notions: *sender-receiver anonymity* and *unobservability*.

5.1 Broadcast Networks

Broadcast channels can be used as a straightforward approach to obtain some form of receiver anonymity [48]. In general, the most obvious protocol of transmitting a message over the broadcast channel is trivially RA-anonymous. Blaze et al. [8] recently suggested a protocol for anonymous routing in the context of wireless networks. Very roughly, their basic protocol is an adaptation of onion routing [29] to broadcast networks. The operation of sending a message is then analogous, and involves computing a path of routers, and a corresponding *onion* (a nested encryption) of the message (see [8] for details). The difference is that each transmission of the "onion" between routers is done via the broadcast channel, so all receivers attempt to decrypt the onion but only the intended recipient succeeds (although not mentioned, some integrity mechanism must be used in the onion). Under passive global adversaries, if the encryption used provides key-privacy [4],[13] the protocol can easily be shown RA*-anonymous. However, due to the shared nature of the wireless medium, transforming it into a UO-secure protocol may not be practical given the message overhead (unavoidable by Proposition 4.9).

5.2 DC-nets or Anonymous Broadcast

DC-nets [15, 32] can be seen as particular instances of anonymous broadcast protocols [54]. In these protocols, there is a single message sent which is public. In [32], Golle and Juels proposed very efficient anonymous broadcast protocol based on pairings. Whenever a transmission is to take place, all parties participate in the protocol by transmitting "pads". Each pad contains the (potentially empty) message the party intends to transmit. Golle and Juels show how to combine the pads so the transmitted messages are recovered with high probability (and therefore theirs is a message-transmission protocol with high probability). They also show how each party can provide a non-interactive zero-knowledge (NIZK) proof [21] for the correctness of her pad without revealing the underlying message. By the simulatability of the NIZK proof, it then follows that their protocol can be proven SA-anonymous under global passive adversaries as long as the *Bilinear Diffie-Hellman assumption* [9] holds. Notice that this result is not implied by their security proof as the anonymity notion used in [32] is arguably different (see Section 1.4).

5.3 MIX networks:

Robust and efficient MIX-net constructions can be built from efficient schemes to *prove a shuffle* [25, 33, 44]. In these constructions, each mixer proves the correctness of the shuffle operation (usually a random permutation and sometimes partial decryption) was done correctly. The resulting mix-net protocol may work as follows: first, all senders send encryptions of their messages to the first mixer (the encryptions are made under a threshold key shared by the mixers). Then, the mixing process starts where each mixer performs (and proves) her shuffle passing the resulting vector to the next mixer. The last mixer broadcast the resulting vector. The shuffles in [33] and [24, Appendix A] can be proven *honest verifier zero-knowledge* (HVZK) arguments. The shuffles in [25, 44] can be shown to satisfy the stronger property IND-CPA$_S$ [44]. Under passive adversaries, both properties suffice to prove the adversary cannot distinguish two executions of the associated mix-nets even under adversarial inputs. Assuming the last mixer broadcasts the output, these constructions can then be proven RA*-secure.

[13] This requirement apparently was overlooked in [8].

6 Variants and Extensions

k-ANONYMITY: Intuitively, a protocol achieves k-anonymity if any adversary trying to determine the sender (resp. receiver) of a message can only narrow the sender's identity down to no less than k possible senders (resp. receivers). The concept was proposed by Pfitzmann [45] and further developped (along with efficient constructions) by von Ahn et al. [56] as a way to improve the efficiency of DC-nets. We can accommodate the notion of k-anonymity in our framework by further restricting the relation $R_{\mathbf{N}}$. For each of the message matrices output by the adversary we require at least k non-empty rows (resp. columns) to capture the restriction to k senders (resp. receivers).

PASSIVE ADVERSARIES WITH CORRUPTIONS: As mentioned before, it is possible to extend our framework to consider party corruptions. The adversary would be allowed to passively (either statically or dynamically) corrupt senders and receivers, with the obvious restrictions that the local inputs and outputs corresponding to the corrupted parties must be the same in the two message matrices output by the adversary. Note that this conditions immediately hold if the corrupted party that does not send or receive messages and only acts as forwarder (router). The security proofs for the protocols mentioned in previous section carry to this stronger model. Extending our framework beyond passive attacks (active adversaries) is currently part of ongoing research.

References

[1] M. Abe. Universally verifiable mix-net with verification work independent of the number of mix-servers. In K. Nyberg (ed.) *Proc. of EUROCRYPT'98*, volume 1403 of *LNCS*, pages 437–447. Springer–Verlag, 1998.

[2] D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. In B. Awerbuch (ed.) *Proc. of the 22nd Annual ACM Symposium on the Theory of Computing – STOC'90*, pages 503–513. ACM Press. 1990.

[3] A. Beimel and S. Dolev. Buses for anonymous message delivery. *Journal of Cryptology*, 16, 2003.

[4] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In C. Boyd (ed.) *Proc. of ASIACRYPT'2001*, volume 2248 LNCS, pages 566–582. Springer–Verlag, 2001.

[5] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk (ed.) *Proc. of CRYPTO'98, Proceedings*, volume 1462 of *LNCS*, pages 26–45. Springer–Verlag, 1998.

[6] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for noncryptographic fault-tolerant distributed computations. In *Proc. of the 20th Annual ACM Symposium on Theory of Computing*, pages 1–10. ACM Press, 1988.

[7] R. Berman, A. Fiat, and A. Ta-Shma. Provable unlinkability against traffic analysis. In *Proc. Financial Cryptography – FC'04*, vol. 3110 of *LNCS*. Springer, 2004.

[8] M. Blaze, J. Ioannidis, A.D. Keromytis, T. Malkin, and A. Rubin. WAR: Wireless anonymous routing. In *Security Protocols Workshop*, volume 3364 of *LNCS*, pages 218–232. Springer-Verlag, 2003.

[9] D. Boneh and M.K. Franklin. Identity-based encryption from the weil pairing. In *Proc. of CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer–Verlag, 2001.

[10] J. Bos and B. den Boer. Detection of disrupters in the DC protocol. In J.-J. Quisquater and J. Vandewalle (eds.) *Proc. of EUROCRYPT'89*, volume 434 of *LNCS*, pages 320–328. Springer-Verlag, 1989.

[11] J. Camenisch and A. Lysyanskaya. A formal treatment of onion routing. In V. Shoup (ed.) *Proc. of CRYPTO'05*, volume 3621 of *LNCS*, pages 169–187. Springer-Verlag, 2005.

[12] R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.

[13] R. Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE Computer Society Press, 2001.

[14] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[15] D. Chaum. The Dining Cryptographers Problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[16] D. Chaum, C. Crepeau, and I. Damgård. Multiparty unconditional secure protocols. In *Proc. of STOC'88*, pages 11–19. ACM Press, 1988.

[17] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proc. of IEEE Security and Privacy*, 2003.

[18] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In R. Dingledine and P. Syverson (eds.) *Proc. of Privacy Enhancing Technologies Workshop – PET ' 02*, volume 2482 of *LNCS*. Springer-Verlag, 2002.

[19] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proc. of the 13th USENIX Security Symposium*, 2004.

[20] S. Dolev and R. Ostrobsky. Xor-trees for efficient anonymous multicast and reception. *ACM Transactions on Information System Security*, 3(2):63–84, 2000.

[21] U. Feige, D. Lapidot, and A. Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM Journal on Computing*, 29(1), 1999.

[22] J. Feigenbaum, A. Johnson, and P. Syverson. A model for onion routing with provable anonymity. In *Financial Cryptography*, vol. 4886 LNCS, Springer, 2007.

[23] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In J. Seberry and J. Pieprzyk (eds.) *Proc. of AUSCRYPT '92*, volume 718 of *LNCS*, pages 244–251. Springer-Verlag, 1992.

[24] J. Furukawa. Efficient, verifiable shuffle decryption and its requirement of unlinkability. In *Proc. of Practice and Theory in Public Key Cryptography – PKC ' 04*, LNCS. Springer-Verlag, 2004.

[25] J. Furukawa and K. Sako. An efficient scheme for proving a shuffle. In *Advances in Cryptology – CRYPTO ' 2001*, volume 2139 of *LNCS*. Springer-Verlag, 2001.

[26] F.D. Garcia, I. Hasuo, W. Pieters, and P. van Rossum. Provable anonymity. In *Proc. of the 3rd ACM Workshop on Formal Methods in Security Engineering – FMSE'05*, pages 63–72. ACM Press, 2005.

[27] O. Goldreich. A uniform complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[28] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *Proc. 27th Symposium on Foundations of Computer Science*, pages 174–187. IEEE Press, 1986.

[29] D.M. Goldschlag, M.G. Reed, and P.F. Syverson. Hiding Routing Information. In R. Anderson (ed.) *Proc. of Information Hiding*, volume 1174 of *LNCS*, pages 137–150. Springer-Verlag, 1996.

[30] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Science*, 28:270–299, 1984.

[31] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *Siam J. of Computing*, 17(2):281–308, 1988.

[32] P. Golle and A. Juels. Dining cryptographers revisited. In *Proc. of Eurocrypt'04*, volume 3027 of *LNCS*. Springer-Verlag, 2004.

[33] J. Groth. A verifiable secret shuffle of homomorphic encryptions. In *Proc. of Public Key Cryptography – PKC ' 03*, LNCS. Springer-Verlag, 2003.

[34] C. Gülcü and G. Tsudik. Mixing E-mail with Babel. In *Proc. of the Network and Distributed Security Symposium – NDSS '96*, pages 2–16. IEEE Press, 1996.

[35] A. Hevia and D. Micciancio. Indistinguishability-based Characterization of Anonymous Channels, In *Proc. of Privacy Enhancing Technologies Workshop – PET ' 08*, volume 5??? of *LNCS*. Springer-Verlag, 2008.

[36] D. Hughes and V. Shmatikov. Information Hiding, Anonymity and Privacy: a Modular Approach. *Journal of Computer Security*, 12(1):3–36, 2004.

[37] J.Y. Halpern and K.R. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 2004.

[38] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography from anonymity. In *Proc. of FOCS'06*, IEEE Press, 2006.

[39] M. Jakobsson, A. Juels, and R.L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In *Proc. of the 11th USENIX Security Symposium (SECURITY-02)*, pages 339–353, USENIX Association. 2002.

[40] D. Kesdogan, J. Egner, and R. Büschkes. Stop-and-go MIXes: Providing probabilistic anonymity in an open system. In *Proc. of Information Hiding Workshop – IH ' 98*, volume 1525 of *LNCS*. Springer-Verlag, 1998.

[41] S. Mauw, J.H.S. Verschuren, and E.P. de Vink. A formalization of anonymity and onion routing. In *European Symposium on Research in Computer Security – ESORICS ' 04*, LNCS. Springer-Verlag, 2004.

[42] S. Micali, C. Rackoff, and B. Sloan. The notion of security for probabilistic cryptosystems. *Siam Journal of Computing*, 17(2):412–426, 1988.

[43] A. Neff. A verifiable secret shuffle and its application to E-voting. In *Proc. 8th ACM Conference on Computer and Communications Security*. ACM SIGSAC, 2001.

[44] L. Nguyen, R. Safavi-Naini, and K. Kurosawa. Verifiable shuffles: A formal model and a paillier-based efficient construction with provable security. In *Proc. of Applied Cryptography and Network Security (ACNS'04)*, volume 3089 *LNCS*. Springer–Verlag, 2004.

[45] A. Pfitzmann How to Implement ISDNs Without User Observability – some Remarks. Tech. report Fakultät für Informatik, Universität Karlsruhe, 1985.

[46] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity — A proposal for terminology. *LNCS*, 2009:1–9, 2001.

[47] A. Pfitzmann, B. Pfitzmann, and M. Waidner. ISDN-Mixes: Untraceable communication with very small bandwidth overhead. Proc. Kommunikation in verteilten Systemen, Informatik-Fachberichte 267, 451-463. Springer-Verlag, 1991. Slightly extended in: Information Security, Proc. IFIP/Sec'91, 245–258, 1991.

[48] A. Pfitzmann and M. Waidner. Networks without user observability. *Computers & Security*, 6(2):158–166, 1987.

[49] C. Rackoff and D. R. Simon. Cryptographic defense against traffic analysis. In *Proc. of STOC'93*, pages 672–681, ACM Press, 1993.

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

[51] M. Rennhard and B. Plattner. Practical anonymity for the masses with morphmix. In Ari Juels (ed.) *Proc. of Financial Cryptography – FC '04*, volume 3110 of *LNCS*. Springer-Verlag, 2004.

[52] A. Serjantov. *On the Anonymity of Anonymity Systems*. PhD thesis, University of Cambridge, 2004.

[53] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In R. Dingledine and P. Syverson (ed.) *Proc. of Privacy Enhancing Technologies Workshop – PET ' 02*, volume 2482 of *LNCS*. Springer-Verlag, 2002.

[54] F. Stajano and R. Anderson. The cocaine auction protocol: On the power of anonymous broadcast. In Andreas Pfitzmann, editor, *Information Hiding —3rd International Workshop, IH'99*, volume 1768 of *LNCS*. Springer-Verlag, 2000.

[55] P.F. Syverson and S.G. Stubblebine. Group principals and the formalization of anonymity. In *Proc. of the World Congress on Formal Methods*, volume 1708 of *LNCS*, pages 814–833. Springer–Verlag, 1999.

[56] L. von Ahn, A. Bortz, and N.J. Hopper. k-Anonymous message transmission. In V. Atluri and P. Liu (ed.) *Proc. of the 10th ACM Conference on Computer and Communication Security – CCS'03*, pages 122–130, ACM Press. 2003.

[57] M. Waidner. Unconditional sender and recipient untraceability in spite of active attacks. In J.-J. Quisquater and J. Vandewalle (eds.) *Proc. of EUROCRYPT' 89*, volume 434 *LNCS*, pages 302–319. Springer-Verlag, 1990.

[58] M. Waidner and B. Pfitzmann. The dining cryptographers in the disco: Unconditional sender and recipient untraceability with computationally secure serviceability. In J.-J. Quisquater and J. Vandewalle (eds.) *Proc. of EUROCRYPT'89*, volume 434 *LNCS*, page 690. Springer–Verlag, 1989.

[59] D. Wikström. A universally composable mix-net. In *Theory of Cryptography TCC'04*, volume 2951 *LNCS*, pages 317–335. Springer-Verlag, 2004.

A Public Key Infrastructure and Key-Private Encryption

PUBLIC-KEY INFRASTRUCTURE (PKI): In the PKI model, we assume all parties $P_1, \ldots, P_n$ hold the same vector $pk_1, \ldots, pk_n$ of public keys for a certain encryption scheme, and each party P_i holds a secret key sk_i corresponding to pk_i. We assume that the pair (pk_i, sk_i) was correctly generated for each (honest) party P_i.

KEY-PRIVATE ASYMMETRIC ENCRYPTION [4]: Let $\mathcal{AS} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme.[14] Let $b \in \{0,1\}$, $k \in \mathbb{N}$ be the security parameter. Consider the following experiment:

Experiment $\mathbf{Exp}^{ik-cpa-b}_{\mathcal{AS},A}(k)$
  $(pk_0, sk_0) \stackrel{R}{\leftarrow} \mathcal{K}(1^k)$, $(pk_1, sk_1) \stackrel{R}{\leftarrow} \mathcal{K}(1^k)$
  $(x, s) \leftarrow A(\text{“find”}, pk_0, pk_1)$
  $y \leftarrow \mathcal{E}_{pk_b}(x)$
  $g \leftarrow A(\text{“guess”}, y, s)$
  return g

An encryption scheme $\mathcal{AS}$ achieves key privacy against chosen plaintext attack (IK-CPA) if the quantity

$$\mathbf{Adv}^{ik-cpa}_{\mathcal{AS},A}(k) \stackrel{\text{def}}{=} \Pr\left[\mathbf{Exp}^{ik-cpa-1}_{\mathcal{AS},A}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{ik-cpa-0}_{\mathcal{AS},A}(k) = 1\right]$$

is negligible in k for any feasible (PPT in k) adversary A.

[14] For simplicity, we assume any common parameters for the encryption scheme are generated initially once and for all.

B Examples of Hidden Communication Patterns

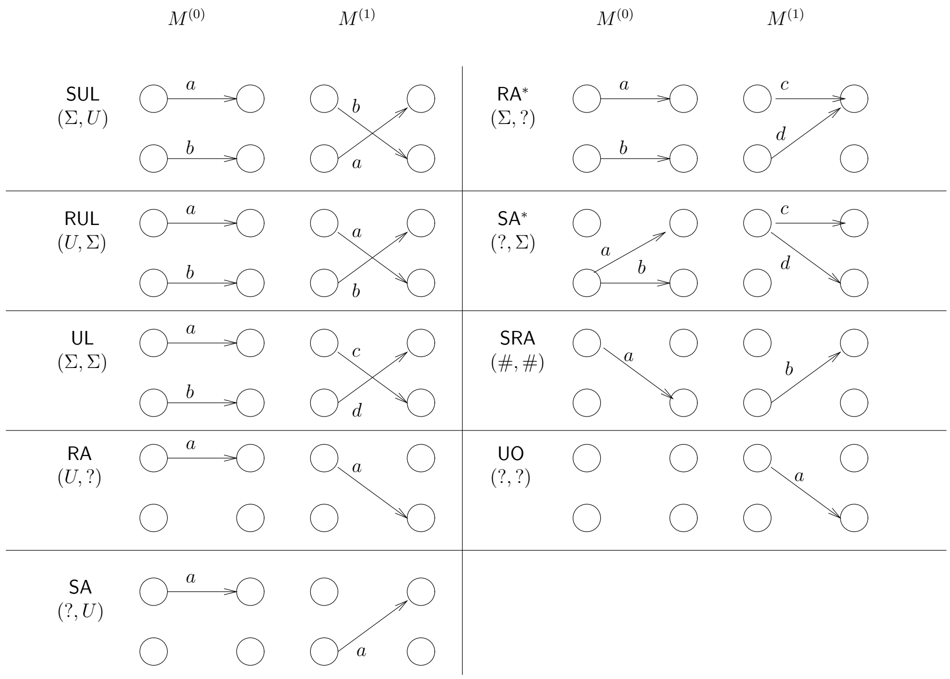

Figure 3: A pictorial representation of toy examples of communication patterns hidden by each anonymity notion. For each notion, there are two communication patterns illustrated by graphs of four nodes: the leftmost graph represents the communication pattern for the combination of senders, messages, and receivers corresponding to matrix $M^{(0)}$, while the rightmost graph the pattern specified by $M^{(1)}$. For each graph the nodes which represent parties, arrows represent messages, and the label is the message value; the nodes where arrows depart represent senders, and those where arrows arrive represent receivers.