Christiane Kuhn*, Martin Beck, Stefan Schiffner, Eduard Jorswieck, and Thorsten Strufe

On Privacy Notions in Anonymous Communication

Abstract: Many anonymous communication networks (ACNs) with different privacy goals have been developed. Still, there are no accepted formal definitions of privacy goals, and ACNs often define their goals ad hoc. However, the formal definition of privacy goals benefits the understanding and comparison of different flavors of privacy and, as a result, the improvement of ACNs. In this paper, we work towards defining and comparing privacy goals by formalizing them as privacy notions and identifying their building blocks. For any pair of notions we prove whether one is strictly stronger, and, if so, which. Hence, we are able to present a complete hierarchy. Using this rigorous comparison between notions, we revise inconsistencies between the existing works and improve the understanding of privacy goals.

Keywords: Anonymity, Privacy notion, Anonymous Communication, Network Security

1 Introduction

With our frequent internet usage of, e.g., social networks, instant messaging, and web browsing, we constantly reveal personal data. Content encryption can reduce the footprint, but metadata (e.g. correspondents' identities) still leaks. To protect metadata from state and industrial surveillance, a broad variety of anonymous communication networks (ACNs) has emerged; one of the most deployed is Tor [8], but also others, e.g. I2P [17] or Freenet [7], are readily avail-

able. Additionally, many conceptual systems, like Mix-Nets [6], DC-Nets [4], Loopix [15] and Crowds [16] have been published.

The published ACNs address a variety of privacy goals. However, many definitions of privacy goals are ad hoc and created for a particular use case. We believe that a solid foundation for future analysis is still missing. This hinders the understanding and comparison of different privacy goals and, as a result, comparison and improvement of ACNs. In general, comparing privacy goals is difficult since their formalization is often incompatible and their naming confusing. This has contributed to a situation where existing informal comparisons disagree: e.g., Sender Unlinkablity of Hevia and Micciancio's framework [12] and Sender Anonymity of AnoA [1] are both claimed to be equivalent to Sender Anonymity of Pfitzmann and Hansen's terminology [14], but significantly differ in the protection they actually provide. These naming issues further complicate understanding of privacy goals and hence analysis of ACNs.

To allow rigorous analysis, i.e. provable privacy, of ACNs, their goals need to be unambiguously defined. Similar to the notions of semantic security (like CPA, CCA1, CCA2 [2]) for confidentiality, privacy goals can be formally defined as indistinguishability games. We call such formally defined privacy goals *privacy notions*. Further, notions need to be compared according to their strength: achieving the stronger notion implies the weaker one. Comparison of notions, and of the ACNs achieving them, is otherwise impossible. To understand the ramifications of privacy goals, we aim at setting all notions into mutual relationships. This means for every pair of notions it must be clear if one is stronger or weaker than the other, or if they have no direct relationship. Such a comparison has already been made for the notions of semantic security [2].

In this work, we tackle the formal definition and comparison of privacy goals. To achieve this, we build on the foundations of existing analytical frameworks [1, 3, 10, 12]. With their preparatory work, we are able to present basic building blocks of privacy notions: observable properties of a communication, that (depending on the notion) must either be protected, i.e. kept private, by the protocol, or are permitted to be learned by the adversary. Defining our notions based on the idea of

*Corresponding Author: Christiane Kuhn: TU Dresden, E-mail: christiane.kuhn@tu-dresden.de
Martin Beck: TU Dresden, E-mail: martin.beck1@tu-dresden.de
Stefan Schiffner: Université du Luxembourg, E-mail: stefan.schiffner@uni.lu
Eduard Jorswieck: TU Dresden, E-mail: eduard.jorswieck@tu-dresden.de
Thorsten Strufe: TU Dresden, E-mail: thorsten.strufe@tu-dresden.de

properties simplifies comparison. Further, we map practitioners' intuitions to their underlying formal model, justify our choice of notions with exemple use cases for each, and make a sanity check to see that the privacy goals of a current ACN (Loopix [15]) are covered. Additionally, for all formalized goals of existing analysis frameworks [1, 3, 10, 12] we reason to which notions they correspond if they are broken down to the general observable properties and interpreted for ACNs. This means that we focus on general privacy goals and do not present aspects regarding the adversary model, infrequently-used observable information, or the quantification of privacy goals. However, those aspects are compatible with our formalization and have not been ignored; they are presented in the long version of this paper [13].

We compare all identified privacy notions and present a complete proven hierarchy. As a consequence of our comparison, we are able to rectify mapping inconsistencies of previous work and show how privacy notions and data confidentiality interact. Furthermore, the proofs for building the hierarchy include templates in order to compare and add new privacy notions to the established hierarchy, if necessary.

In summary, our main contributions are:
– the mapping of practitioners' intuitions to game-based proofs,
– the definition of building blocks for privacy notions,
– the selection and unified definition of notions,
– a complete hierarchy of privacy notions, which simplifies comparison of ACNs, and
– the resolution of inconsistencies and revision of mistakes in previous (frame)works.

Outline. Section 2 contains an introductory example and gives an overview of our paper. In Section 3, we introduce the underlying model and indistinguishability games. In Section 4, we introduce the basic building blocks of privacy notions: properties. In Section 5, we define the privacy notions. In Section 6, we argue our choice of notions. In Section 7, we present the relations between the notions. In Section 8, we discuss our results. In Section 9, we conclude our paper and give an outlook.

2 Overview

We start with an example of a use case and the corresponding implicit privacy goal, to then introduce the idea of the related indistinguishability game. We show how such a game works and what it means for a protocol to be secure according to this goal. Furthermore, by

adopting the game we sketch how privacy goals can be formalized as notions and provide an intuition for the relations between different goals.

EXAMPLE: *Alice is a citizen of a repressive regime and engaged with a resistance group. Despite the regime's sanctions on distributing critical content, Alice wants to publish her latest critical findings.* A vanilla encryption scheme would reduce Alice's potential audience and thus does not solve her problem. Hence, she needs to hide the link between the message and herself as the sender. We call this goal sender-message unlinkability.[1]

First attempt. We start by presenting an easy game, that at first glance looks like the correct formalization for the goal of the example, but turns out to model an even stronger goal.

For Alice's safety, the regime should not suspect her of being the sender of a compromising message, otherwise she risks persecution. Thus, we need to show for the applied protection measure, that compared to any other sender of this message, it is not more probable that Alice is the sender. We analyze the worst case: in a group of users, let Charlie be a user for whom the probability of being the sender differs most from Alice's probability. If even these two are too close to distinguish, Alice is safe, since all other probabilities are closer. Hence, the regime cannot even exclude a single user from its suspects.

We abstract this idea into a game[2], where the adversary aims to distinguish two "worlds" or scenarios. These may only differ in the properties the protocol is required to protect, but within these restrictions the adversary can choose freely, especially the worst case that is easiest for her to distinguish (e.g. in one scenario Alice sends the message, in the other Charlie). Fig. 1 shows such a game.

What the adversary can observe in step 4 depends on her capabilities and area of control. A weak adversary may only receive a message from somewhere, or discover it on a bulletin board. However, a stronger adversary could e.g. also observe the activity on the Internet uplinks of some parties.

The adversary wins the game if she guesses the correct scenario. If she can devise a strategy that allows her

---

1 Usually this is called sender anonymity. However, since the term sender anonymity is overloaded and sometimes also used with a slightly different meaning, we refer to it as sender-message unlinkability, as the message should not be linkable to the sender.
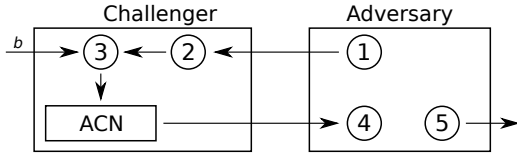**2** Similar to indistinguishability games in cryptology [11].

Fig. 1. Steps of the sample game: **1)** adversary picks two scenarios; **2)** challenger checks if scenarios only differ in senders; **3)** based on random bit b the challenger inputs a scenario into the ACN; **4)** adversary observes execution; **5)** adversary outputs 'guess' as to which scenario was executed

to win the game repeatedly with a probability higher than random guessing, she must have learned some information that is supposed to be protected, here the sender (e.g. that Alice is more probable the sender of the message than Charlie), since everything else was identical in both scenarios. Hence, we say that, if the adversary can find such a strategy, we do not consider the analyzed protocol secure regarding the respective privacy goal.

Why this is too strong. As argued, a protocol achieving this goal would help Alice in her use case. However, if an adversary learns who is sending any message with real information (i.e. no random bits/dummy traffic), she can distinguish both scenarios and wins the game. As an example, consider the following two scenarios: (1) Alice and Bob send messages (2) Charlie and Dave send messages. If the adversary can learn the active senders, she can distinguish the scenarios and win the game. However, if she only learns the set of active senders, she may still not know who of the two active senders in the played scenario actually sent the regime-critical content. Thus, a protocol hiding the information of who sent a message within a set of active senders is good enough for the given example. Yet, it is considered insecure regarding the above game, since an adversary can learn the active senders. Hence, the game defines a goal stronger than the required sender-message unlinkability. As the ACN in this case needs to hide the sending activity (the adversary does not know if a certain possible sender was active or not), we call the goal that is actually modeled sender unobservability.

Correcting the formalization. However, we can adjust the game of Fig. 1 to model sender-message unlinkability. We desire that the only information about the communications that differs between the scenarios is who is sending which message. Thus, we allow the adversary to pick scenarios that differ in the senders, but not in the activity of the senders, i.e. the number of messages each active sender sends. This means, we change what the adversary is allowed to submit in step

1 and what the challenger checks in step 2. So, if the adversary now wants to use Alice and Charlie, she has to use both in both scenarios, e.g. (1) Alice sends the critical message, Charlie a benign message and (2) Charlie sends the critical message, Alice the benign message. Hence, given an ACN where this game cannot be won, the adversary is not able to distinguish whether Alice or another active user sent the regime-critical message. The adversary might learn, e.g. that someone sent a regime-critical message and the identities of all active senders (here that Alice and Charlie are active senders). However, since none of this is sanctioned in the above example, Alice is safe, and we say such an ACN provides sender-message unlinkability.

Lessons learned. Depending on the formalized privacy goal (e.g. sender unobservability) the scenarios are allowed to differ in certain properties of the communications (e.g. the active senders) as we have illustrated in two exemple games. Following the standard in cryptology, we use the term *privacy notion*, to describe such a formalized privacy goal that defines properties to be hidden from the adversary.

Further, the games used to prove the privacy notions only differ in how scenarios can be chosen by the adversary and hence what is checked by the challenger. This also holds for all other privacy notions; they all define certain properties of the communication to be private and other properties that can leak to the adversary. Therefore, their respective games are structurally identical and can be abstracted to define one general game, whose instantiations represent notions. We explain and define this general game in Section 3. We then define the properties (e.g. that the set of active senders can change) in Section 4 and build notions (e.g. for sender unobservability) upon them in Section 5.

Additionally, we already presented the intuition that sender unobservability is stronger than sender-message unlinkability. This is not only true for this example, in fact we prove: every protocol achieving sender unobservability also achieves sender-message unlinkability. Intuitively, if whether Alice is an active sender or not is hidden, whether she sent a certain message or not is also hidden. We will prove relations between our privacy notions in Section 7 and show that the presented relations (depicted in Figure 3) are complete. Before that, we argue our choice of notions in Section 6.

3 Our Game model

Our goal in formalizing the notions as a game is to analyze a given ACN protocol w.r.t. to a notion, i.e. the game is a tool to investigate if an adversary can distinguish two self-chosen, notion-compliant scenarios. Scenarios are sequences of communications. A *communication* is described by its sender, receiver, message and auxiliary information (e.g. session identifiers) or the empty communication, signaling that nobody wants to communicate at this point. Some protocols might restrict the information flow to the adversary to only happen at specific points in the execution of the protocol, e.g. because a component of the ACN processes a batch of communications before it outputs statistics about them. Therefore, we introduce *batches* as a sequence of communications, which is processed as a unit before the adversary observes anything[3]. When this is not needed, batches can always be replaced with single communications.

As explained in Section 2, we do not need to define a complete new game for every privacy goal, since notions only vary in the difference between the alternative scenarios chosen by the adversary. Hence, for a given ACN and notion, our general game is simply instantiated with a model of the ACN, which we call the protocol model, and the notion. The protocol model accepts a sequence of communications as input. Similar to the real implementations the outputs of the protocol model are the observations the real adversary can make. Note, the adversaries in the game and the real world have the same capabilities[4], but differ in their aims: while the real world adversary aims to find out something about the users of the system, the game adversary merely aims to distinguish the two scenarios she has constructed herself.

In the simplest version of the game, the adversary constructs two scenarios, which are just two batches of communications and sends them to the challenger. The challenger checks that the batches are compliant with the notion. If so, the challenger tosses a fair coin to randomly decide which of the two batches it executes with the protocol model. The protocol model's output is returned to the game adversary. Based on this infor-

mation, the game adversary makes a guess about the outcome of the coin toss.

We extend this simple version of the game, to allow the game adversary to send multiple times two batches to the challenger. However, the challenger performs a single coin flip and sticks to this scenario for this game, i.e. it always selects the batches corresponding to the initial coin flip. This allows analyzing for adversaries, that are able to base their next actions in the attack on the observations they made previously.

To unfetter our general game from the concrete adversary model, we allow the adversary to send protocol queries. This is only a theoretical formalization to reflect what information the adversary gets and what influence she can exercise. These protocol query messages are sent to the protocol model without any changes by the challenger. The protocol model covers the adversary to ensure that everything the real world adversary can do is possible in the game with some query message. For example, protocol query messages can be used to add or remove nodes from the ACN by sending the appropriate message.

As introduced in Section 2, we say that an adversary has an advantage in winning the game, if she guesses the challenger-selected scenario correctly with a higher probability than random guessing. A protocol achieves a certain privacy goal, if an adversary has at most negligible advantages in winning the game.

Formalization

In this subsection, we formalize the game model to conform to the above explanation.

We use Π to denote the analyzed *ACN protocol model*, Ch for the challenger and $\mathcal{A}$ for the adversary, which is a probabilistic polynomial time algorithm. Additionally, we use X as a placeholder for the specific notion, e.g. sender unobservability, if we explain or define something for all the notions. A *communication* r in Π is represented by a tuple (u, u', m, aux) with a sender u, a receiver u', a message m, and auxiliary information aux (e.g. session identifiers). Further, we use $\Diamond$ instead of the communication tuple (u, u', m, aux) to represent that no communication occurs. Communications are clustered into *batches* $\underline{r}_b = (r_{b_1}, \ldots, r_{b_l})$, with r_{b_i} being the i-th communication of batch $\underline{r}_b$. Note that we use $\underline{r}$ (underlined) to identify batches and r (no underline) for single communications. Batches in turn are clustered into *scenarios*; the first scenario is $(\underline{r}_{0_1}, \ldots, \underline{r}_{0_k})$. A *challenge* is defined as the tuple of

---

3 We use the word batch to designate a bunch of communications. Besides this similarity, it is not related to batch mixes.
4 A stronger game adversary also implies that the protocol is safer in the real world.

two scenarios $\big((\underline{r}_{0_1}, \ldots, \underline{r}_{0_k}), (\underline{r}_{1_1}, \ldots, \underline{r}_{1_k})\big)$. All symbols used so far and those introduced later are summarized in Tables 5 – 7 in Appendix D.

Simple Game.

1. Ch randomly picks challenge bit b.
2. $\mathcal{A}$ sends a batch query, containing $\underline{r}_0$ and $\underline{r}_1$, to Ch.
3. Ch checks if the query is valid, i.e. both batches differ only in information that is supposed to be protected according to the analyzed notion X.
4. If the query is valid, Ch inputs the batch corresponding to b to Π.
5. Π's output $\Pi(\underline{r}_b)$ is handed to $\mathcal{A}$.
6. After processing the information, $\mathcal{A}$ outputs her guess g for b.

Extensions. As explained above, there are useful extensions we make to the simple game:

Multiple Batches Steps 2-5 can be repeated.

Other parts of the adversary model Instead of Step 2, $\mathcal{A}$ can also decide to issue a protocol query, containing an input specific to Π and receive Π's output to it (e.g. the internal state of a router that is corrupted in this moment). This might change Π's state.

Achieving notion X. Intuitively, a protocol Π achieves a notion X if any possible adversary has at most negligible advantage in winning the game. To formalize the informal understanding of Π achieving goal X, we need the following denotation. $\Pr[g = \langle \mathcal{A} \mid Ch(\Pi, X, b)\rangle]$ describes the probability that $\mathcal{A}$ outputs g, when Ch is instantiated with Π and X and the challenge bit was chosen to be b. With this probability, achieving a notion translates to Definition 1.

Definition 1 (Achieving a notion X). *An ACN Protocol Π achieves X, iff for all probabilistic polynomial time(PPT) algorithms $\mathcal{A}$ there exists a negligible δ such that*

$$\Big| Pr[0 = \langle \mathcal{A} \mid Ch(\Pi, X, 0)\rangle] - Pr[0 = \langle \mathcal{A} \mid Ch(\Pi, X, 1)\rangle] \Big| \leq \delta.$$

4 Protected Properties

We define properties to specify which information about the communication is allowed to be disclosed to the adversary, and which must be protected to achieve a privacy notion, as mentioned in Section 2. We distinguish between simple and complex properties. Simple properties can be defined with the basic game model already

Symbol	Description	Translation to Game
$|M|$	Message Length	Messages in the two scenarios always have the same length.
E_S	Everything but Senders	Everything except the senders is identical in both scenarios.
E_R/E_M	Everything but Receivers/Messages	Analogous
E_{SM}	Everything but Senders and Messages	Everything except the senders and messages is identical in both scenarios.
E_{RM}/E_{SR}	Analogous	Analogous
$\lozenge\!\!\!/$	Something is sent	In every communication something must be sent ($\lozenge$ not allowed).
$\aleph$	Nothing	Nothing will be checked; always true.
U/U'	Active Senders/Receivers	Who sends/receives is equal for both scenarios.
Q/Q'	Sender/Receiver Frequencies	Which sender/receiver sends/receives how often is equal for both scenarios.
$|U|/|U'|$	Number of Senders/ Receivers	How many senders/receivers communicate is equal for both scenarios.
P/P'	Message Partitioning per Sender/Receiver	Which messages are sent/received from the same sender/receiver is equal for both scenarios.
H/H'	Sender/Receiver Frequency Histograms	How many senders/receivers send/receive how often is equal for both scenarios.

Table 1. Simple properties; information about communications that may be required to remain private

introduced, while complex properties require some extensions to the basic model.

4.1 Simple Properties

We summarize the informal meaning of all simple properties in Table 1 and introduce them in this section.

Assume an ACN aims to hide the sender but discloses message lengths to observers. For this case, we specify the property $(|M|)$ that the message length must not differ between the two scenarios, as this information must not help the adversary to distinguish which scenario the challenger chose to play.

Next, we might want an ACN to protect the identity of a sender, as well as any information about who sent a message, but deliberately disclose which messages are received by which receiver, who the receivers are, and potentially other auxiliary information. We hence specify a property (E_S) where only the senders differ between the two scenarios[5], to ensure that the adversary in our game can only win by identifying senders. In case the protection of the receiver identities or messages is required, the same can be defined for receivers (E_R) or messages (E_M).

Further, we might want the ACN to protect senders and also the messages; leaving the receiver and auxil-

---

5 E symbolizes that only this property may vary in the two submitted scenarios and everything else remains equal.

iary information to be disclosed to the adversary. This is achieved by specifying a property where only senders and messages differ between the two scenarios and everything else remains equal (E_{SM}). Again, the same can be specified for receivers and messages (E_{RM}) or senders and receivers (E_{SR}).

Lastly, ACNs might allow the adversary to learn whether a real message is sent or even how many messages are sent. We specify a property ($\emptyset$) that requires real communications in both scenarios, i.e. it never happens that nothing is sent in one scenario but something is sent in the other. We ensure this by not allowing the empty communication ($\diamond$).

However, a very ambitious privacy goal might even require that the adversary learns no information about the communication at all ($\aleph$). In this case, we allow any two scenarios and check nothing.

Formalizing those Simple Properties. In the following definition all simple properties mentioned so far are formally defined. Therefore, we use $\top$ as symbol for the statement that is always true.

Definition 2 (Properties $|M|$, E_S, E_{SM}, $\emptyset$, $\aleph$). *Let the checked batches be $\underline{r_0}, \underline{r_1}$, which include the communications $r_{0j} \in \{(u_{0_j}, u'_{0_j}, m_{0_j}, aux_{0_j}), \diamond\}$ and $r_{1j} \in \{(u_{1_j}, u'_{1_j}, m_{1_j}, aux_{1_j}), \diamond\}$ with $j \in \{1, \ldots l\}$. We say the following properties are met, iff for all $j \in \{1, \ldots l\}$:*

$$|M| : |m_{0j}| = |m_{1j}|$$

$$E_S : r_{1j} = (\mathbf{u_{1_j}}, u'_{0_j}, m_{0_j}, aux_{0_j})$$

$$E_R : r_{1j} = (u_{0_j}, \mathbf{u'_{1_j}}, m_{0_j}, aux_{0_j})$$

$$E_M : r_{1j} = (u_{0_j}, u'_{0_j}, \mathbf{m_{1_j}}, aux_{0_j})$$

$$E_{SM} : r_{1j} = (\mathbf{u_{1_j}}, u'_{0_j}, \mathbf{m_{1_j}}, aux_{0_j})$$

$$E_{RM} : r_{1j} = (u_{0_j}, \mathbf{u'_{1_j}}, \mathbf{m_{1_j}}, aux_{0_j})$$

$$E_{SR} : r_{1j} = (\mathbf{u_{1_j}}, \mathbf{u'_{1_j}}, m_{0_j}, aux_{0_j})$$

$$\emptyset : \diamond \notin \underline{r_0} \wedge \diamond \notin \underline{r_1}$$

$$\aleph : \top$$

More Simple Properties: Active Users, Frequencies. The properties of Definition 2 are important to formalize privacy, but are by themselves not sufficient. Take the ACN Tor as an example: While the set of active senders is trivially known to their ISPs and the guard nodes, we still require that the senders are unlinkable with the messages they are sending (and their

receivers). Similarly, the sending (receiving) frequency of a party may be important and is not formalized yet. To formalize these properties, we use sets that capture which user sent which messages in a certain period, i.e. a batch of communications (and similarly sets to capture which user received which messages). Note that we use primes ($'$) for the corresponding sets and properties of the receivers.

Definition 3 (Sender-Message Linking). *We define the sender-message linkings for scenario b (L'_{b_i} the receiver-message linkings are analogous) as:*

$$L_{b_i} := \{(u, \{m_1, ..., m_h\}) \mid u \text{ sent messages } m_1, \ldots, m_h \text{ in batch } i\}.$$

The sets from Definition 3 allow easy identification of who an active sender in this batch was and how often each sent something:

Definition 4 (Active Sender Set, Frequency Set). *Let the current batch be the k-th one. For $b \in \{0, 1\}$ U_b, Q_b (U'_b, Q'_b for L'_b) are defined as:*

$$U_b := \{u \mid (u, M) \in L_{b_k}\}$$

$$Q_b := \{(u, n) \mid (u, M) \in L_{b_k} \wedge |M| = n\}$$

Recall that we currently define properties for ACNs that allow the adversary to learn which senders are active at different times, or the number of messages they send during some periods, while hiding some other properties (e.g. which messages they have sent). Hence, with the respective sets for active users and user frequencies defined, we need only to request that they are equal in both scenarios:

Definition 5 (Properties U, Q, $|U|$). *We say that the properties $U, Q, |U|$ ($U', Q', |U'|$ analogous) are met, iff:*

$$U : U_0 = U_1 \qquad Q : Q_0 = Q_1 \qquad |U| : |U_0| = |U_1|$$

More Simple Properties: Message Partitions, Histograms. Other interesting properties are which messages came from a given sender and how many senders sent how many messages. If the adversary knows which messages are sent from the same sender, e.g. because of a pseudonym, she might be able to combine information from them all to identify the sender. If she knows how many senders sent how many messages, she

knows the sender activity and hence can make conclusions about the nature of the senders.

As before, we introduce auxiliary variables to formally define these two properties. We use $M_{b,I}$ to denote the collection of messages that has been sent by the same sender (e.g. linked by a shared pseudonym) in a set of batches, and $M_{b,I,n}$ to denote the union of all these sets of cardinality n. The equality of the properties in the two scenarios must pertain throughout all comparable batches in the scenarios. If this were not true, the inequality would help the adversary to distinguish the scenarios without learning the protected information e.g. identifying the sender.

Definition 6 (Multi-Batch-Message Linkings). *Let the current batch be the k-th, $\mathcal{K} := \{1, \ldots, k\}$, $\mathcal{P}(\mathcal{K})$ the power set of $\mathcal{K}$ and $\mathcal{U}$ the set of all possible senders ($\mathcal{U}'$ receivers). For $b \in \{0, 1\}$ and $I \in \mathcal{P}(\mathcal{K})$: We define ($M'_{b,I}, M'_{b,I,n}$ for L'_{b_i})*
- *the multi-batch-message-sender linking:*
  $M_{b,I} := \cup_{u \in \mathcal{U}} \{\cup_{i \in I} \{M | (u, M) \in L_{b_i}\}\}$ *and*
- *the cardinality restricted multi-batch-message-sender linking: $M_{b,I,n} := \{M \in M_{b,I} \mid |M| = n\}$.*

As before, we define auxiliary variables capturing the information that we want to be equal in both scenarios: We define ordered sets specifying which messages are sent from the same user for any set of batches (Message Partition P_b) and how many users sent how many messages for any set of batches (Histogram H_b). Therefore, we use a slightly unusual notation: For any set Z, we use $(Z_i)_{i \in \{1, \ldots, k\}}$ to denote the sequence $(Z_1, Z_2, \ldots, Z_k)$ and $\overrightarrow{\mathcal{P}}(Z)$ to denote a sorted sequence of the elements of the power set[6] of Z.

Definition 7 (Message partitions, Histograms). *Consider the k-th batch, $\mathcal{K} := \{1, \ldots, k\}$. For $b \in \{0, 1\}$ P_b, H_b (P'_b, H'_b analogous) are defined as:*

$$P_b := (M_{b,I})_{I \in \overrightarrow{\mathcal{P}}(\mathcal{K})}$$
$$H_b := (\{(n, i) \mid i = |M_{b,I,n}|\})_{I \in \overrightarrow{\mathcal{P}}(\mathcal{K})}$$

Further, we say that properties P, H (P', H' analogous) are met, iff:

$$P : P_0 = P_1 \qquad H : H_0 = H_1$$

4.2 Complex Properties

So far, we have defined various properties to protect senders, messages, receivers, their activity, frequency and the grouping of messages. However, this is not sufficient to formalize several relevant privacy goals, and we must hence introduce complex properties.

Learning Sender and Receiver. Consider that one aims to hide which sender is communicating with which receiver. Early ACNs like classical Mix-Nets [6], and also Tor [8], already used this goal. Therefore, we want the adversary to win the game only if she identifies both: sender and receiver of the same communication.

An intuitive solution may be to model this goal by allowing the adversary to pick different senders and receivers (E_{SR}) in both scenarios (see Fig. 2 (a) for an example). This, however, does not actually model the privacy goal: by identifying only the sender or only the receiver of the communication, the game adversary could tell which scenario was chosen by the challenger. We hence must extend the simple properties and introduce scenario *instances* to model dependencies.

SCENARIO INSTANCES. We now require the adversary to give alternative instances for both scenarios (Fig. 2 (b)). The challenger chooses the scenario according to the challenge bit, which is picked randomly for every game, and the instance according to the instance bit, which is picked randomly for every challenge.

Formally, we replace steps 2–5 of the game with the following steps:

2. $\mathcal{A}$ sends a batch query, containing $\underline{r}_0^0$, $\underline{r}_0^1$, $\underline{r}_1^0$ and $\underline{r}_1^1$ to Ch.

3. Ch checks if the query is valid according to the analyzed notion X.

4. If the query is valid and Ch has not already picked an instance bit a for this challenge, Ch picks $a \in \{0, 1\}$ randomly and independent of b. Then it inputs the batch corresponding to b and a to Π.

5. Π's output $\Pi(\underline{r}_b^a)$ is forwarded to $\mathcal{A}$.

This allows us to model the goal that the adversary is not allowed to learn the sender and receiver: We allow the adversary to pick two sender-receiver pairs, which she uses as instances for the first scenario. The mixed sender-receiver pairs must then be provided as instances for the second scenario (see Fig. 2 (b)). We thus force the game adversary to provide alternative assignments for each scenario. This way she cannot abuse the model to win the game by identifying only the sender or the receiver. We call this property *Random Sender Receiver* R_{SR}.

This complex property is still not sufficient to model the situation in, for example, Tor: The adversary can distinguish the scenarios without learning who sent to whom, just by learning which senders and which receivers are active. Hence, we further restrict the adversary picking instances where both senders and both receivers are active by defining the property *Mix Sender Receiver* M_{SR}. Here, the adversary picks two instances for $b = 0$ where her chosen sender-receiver pairs communicate, and two for $b = 1$ where the mixed sender-receiver pairs communicate. The two instances simply swap the order in which the pairs communicate (Fig. 2 (c)). This way, we force the adversary to provide alternative assignments for each scenario where both suspected senders and both suspected receivers are active. This combination prevents the adversary from winning the game without learning the information that the real system is actually supposed to protect, i.e. the sender-receiver pair.

Fig. 2. Examples showing the general structure of communications that differ in both scenarios: a) Naive, but incorrect b) Random Sender Receiver R_{SR} c) Mixed Sender Receiver M_{SR}

Defining Complex Properties. To simplify the formal definition of complex properties, we introduce *challenge rows*. A challenge row is a pair of communications with the same index that differ in the two scenarios (e.g. r_{0_j}, r_{1_j} with index j). For complex properties, the challenger only checks the differences of the challenge rows in the two scenarios.

Definition 8 (Properties R_{SR}, M_{SR}). *Let the given batches be $\underline{r}_b^a$ for instances $a \in \{0,1\}$ and scenarios $b \in \{0,1\}$, CR the set of challenge row indexes, $(u_0^a, u'^{\,a}_0)$ for both instances $a \in \{0,1\}$ be the sender-receiver-pairs of the first challenge row of the first scenario ($b = 0$). Random Sender Receiver R_{SR}, Mixed Sender Receiver M_{SR} ($R_{SM}, R_{RM}, M_{SM}, M_{RM}$ analogous) are met, iff:*

$$R_{SR}: \quad r_{0\,cr}^a = (\mathbf{u_0^a}, \mathbf{u'^{\,a}_0}, m_{0_{cr}}^1, aux_{0_{cr}}^1) \wedge$$
$$r_{1\,cr}^a = (\mathbf{u_0^a}, \mathbf{u'^{\,1-a}_0}, m_{0_{cr}}^1, aux_{0_{cr}}^1)$$
$$\forall cr \in \mathsf{CR}, a \in \{0,1\}$$

$$M_{SR}: \quad r_{0\,cr}^a = (\mathbf{u_0^a}, \mathbf{u'^{\,a}_0}, m_{0_{cr}}^1, aux_{0_{cr}}^1) \wedge$$
$$r_{0\,cr+1}^a = (\mathbf{u_0^{1-a}}, \mathbf{u'^{\,1-a}_0}, m_{0_{cr}}^1, aux_{0_{cr}}^1) \wedge$$

$$r_{1\,cr}^a = (\mathbf{u_0^a}, \mathbf{u'^{\,1-a}_0}, m_{0_{cr}}^1, aux_{0_{cr}}^1) \wedge$$
$$r_{1\,cr+1}^a = (\mathbf{u_0^{1-a}}, \mathbf{u'^{\,a}_0}, m_{0_{cr}}^1, aux_{0_{cr}}^1)$$
$$\text{for every second } cr \in \mathsf{CR}, a \in \{0,1\}$$

Linking message senders. A final common privacy goal that still cannot be covered is the unlinkability of senders over a pair of messages (Twice Sender Unlinkability). Assume a real world adversary that can determine that the sender of two messages is the same entity. If subsequently she discovers the identity of the sender of one of the messages through a side channel, she can also link the second message to the same individual.

STAGES. To model this goal, we need two scenarios (1) both messages are sent by the same sender, and (2) each message is sent by a different sender. Further, the adversary picks the messages for which she wants to decide whether they are sent from the same individual, and which other messages are sent between those two messages. Therefore, we add the concept of *stages* and ensure that only one sender sends in the challenge rows of stage 1, and in stage 2 either the same sender continues sending ($b = 0$) or another sender sends those messages ($b = 1$). This behavior is specified as the property *Twice Sender T_S.*

Definition 9 (Property T_S). *Let the given batches be $\underline{r}_b^a$ for instances $a \in \{0,1\}$ and scenarios $b \in \{0,1\}$, x the current stage, CR the set of challenge row indexes, $(u_0^a, u'^{\,a}_0)$ for both instances $a \in \{0,1\}$ be the sender-receiver-pairs of the first challenge row of the first scenario ($b = 0$) in stage 1 and $(\tilde{u}_0^a, \tilde{u}'^{\,a}_0)$ the same pairs in stage 2. Twice Sender T_S is met, iff (T_R analogous):*

$$T_S: \quad x = stage1 \, \wedge$$
$$r_{0\,cr}^a = (\mathbf{u_0^a}, u'^{\,0}_0, m_{0_{cr}}^1, aux_{0_{cr}}^1) \wedge$$
$$r_{1\,cr}^a = (\mathbf{u_0^a}, u'^{\,0}_0, m_{0_{cr}}^1, aux_{0_{cr}}^1)$$
$$\vee \quad x = stage2 \, \wedge$$
$$r_{0\,cr}^a = (\mathbf{u_0^a}, \tilde{u}'^{\,0}_0, m_{0_{cr}}^1, aux_{0_{cr}}^1) \wedge$$
$$r_{1\,cr}^a = (\mathbf{u_0^{1-a}}, \tilde{u}'^{\,0}_0, m_{0_{cr}}^1, aux_{0_{cr}}^1)$$
$$\forall cr \in \mathsf{CR}, a \in \{0,1\}$$

Hence, we need to facilitate distinct stages for notions with the complex properties T_S or T_R. Precisely, in step 2 of the game, the adversary is additionally allowed to switch the stages.

Note that the above definition can easily be extended to having more stages and hence, more than two messages for which the adversary needs to decide whether they have originated at the same sender.

This set of properties allows us to specify all privacy goals that have been suggested in literature as privacy notions and additionally all that we consider important. It is of course difficult to claim completeness, as future ACNs may define diverging privacy goals and novel observable properties (or side-channels) may be discovered.

5 Privacy Notions

Given the properties above, we can now set out to express intuitive privacy goals as formal privacy notions. We start by specifying sender unobservability as an example leading to a general definition of our privacy notions.

Recall the first game we defined in Section 2, which corresponds to sender unobservability ($\overline{SO}$ = S(ender) ¬ O(bservability)). There, in both scenarios something has to be sent, i.e. we need to specify that sending nothing is not allowed: $\emptyset$. Further, both scenarios can only differ in the senders, i.e. we also need the property that everything but the senders is equal: E_S. Hence, we define sender unobservability as $\overline{SO} := \emptyset \wedge E_S$. [7]

We define all other notions in the same way:

Definition 10 (Notions). *Privacy notions are defined as a boolean expression of the properties according to Table 2.*

Modeling the notions as a game, the respective challenger will check all aspects of the adversary's queries. A complete description of the challenger can be found in Appendix A.

6 On the Choice of Notions

The space of possible combinations of properties, and hence of conceivable privacy notions, is naturally large. Due to this, we verify our selection of privacy goals by finding exemple use cases. Additionally, we demonstrate the choice and the applicability of our definition by analyzing the privacy goals of Loopix, an ACN that was recently published. We additionally verify that our pri-

---

7 Technically E_S already includes $\emptyset$. However, to make the differences to other notions more clear, we decide to mention both in the definition.

Notion	Properties
$(SR)\overline{L}$	$\emptyset \wedge E_{SR} \wedge M_{SR}$
$(SR)\overline{O}$	$\emptyset \wedge E_{SR} \wedge R_{SR}$
$M\overline{O}$	$\emptyset \wedge E_M$
$M\overline{O} - |M|$	$\emptyset \wedge E_M \wedge |M|$
$M\overline{O}[M\overline{L}]$	$\emptyset \wedge Q \wedge Q'$
$\overline{O}$	$\emptyset$
$C\overline{O}$	$\aleph$
$S\overline{O}$	$\emptyset \wedge E_S$
$S\overline{O} - |U|$	$\emptyset \wedge E_S \wedge |U|$
$S\overline{O} - H$	$\emptyset \wedge E_S \wedge H$
$S\overline{O} - P$	$\emptyset \wedge E_S \wedge P$
$SF\overline{L}$	$\emptyset \wedge E_S \wedge U$
$SF\overline{L} - H$	$\emptyset \wedge E_S \wedge U \wedge H$
$SF\overline{L} - P$	$\emptyset \wedge E_S \wedge U \wedge P$
$SM\overline{L}$	$\emptyset \wedge E_S \wedge Q$
$SM\overline{L} - P$	$\emptyset \wedge E_S \wedge Q \wedge P$
$(2S)\overline{L}$	$\emptyset \wedge E_S \wedge T_S$
$R\overline{O}$ etc.	**analogous**
$S\overline{O}[M\overline{O}]$	$\emptyset \wedge E_{SM}$
$S\overline{O}[M\overline{O} - |M|]$	$\emptyset \wedge E_{SM} \wedge |M|$
$(SM)\overline{O}$	$\emptyset \wedge E_{SM} \wedge R_{SM}$
$(SM)\overline{L}$	$\emptyset \wedge E_{SM} \wedge M_{SM}$
$R\overline{O}[M\overline{O} - |M|]$ etc.	**analogous**
$S\overline{O}\{X'\}$	**Properties of X', remove E_R**
for $X' \in \{R\overline{O}, R\overline{O} - |U'|, R\overline{O} - H', R\overline{O} - P', RF\overline{L}$, $RF\overline{L} - H', RF\overline{L} - P', RM\overline{L}, RM\overline{L} - P'\}$	
$R\overline{O}\{X\}$	**analogous**

Table 2. Definition of the notions. A description of simple properties was given in Table 1.

vacy notions include those of previous publications that suggest frameworks based on indistinguishability games, and provide a complete mapping in Section 6.3.

6.1 Example Use Cases for the Notions

We illustrate our notions by continuing the example of an activist group trying to communicate in a repressive regime, although our notions are generally applicable.

Recall the general idea of an indistinguishability game from the examples in Section 2: To prove that an ACN hides certain properties, whatever is allowed to be learned in the actual ACN must not help a game adversary to win. This way, she is forced to win the game solely based on those properties that are required to remain hidden. Therefore, the information allowed to be disclosed cannot be used in the game and hence must be kept identical in both scenarios.

Before giving examples, we need to order the notions. We chose to group them semantically. Our resulting clusters are shown as gray boxes in Figure 3. Horizontally, we categorize notions that focus on receiver or sender protection (Receiver Privacy Notions or Sender Privacy Notions, respectively) or treat both with the same level of importance (Impartial Notions).

Inside those categories, we use clusters concerning the general leakage type: Both-side Unobservability means that neither senders, nor receivers or messages should be leaked. Both-side Message Unlinkability means that it should be possible to link neither senders nor receivers to messages. In Sender Observability, the sender of every communication can be known, but not the message she sends or to whom she sends (Receiver and Message Observability analogous). In Sender-Message Linkability, who sends which message can be known to the adversary (Receiver-Message and Sender-Receiver Linkability analogous). Table 3 of Appendix D summarizes our naming scheme.

6.1.1 Impartial Privacy Notions

These notions treat senders and receivers equally.

Message Observability. The content of messages can be learned in notions of this group, as messages are not considered confidential. Because the real world adversary can learn the content, we must prevent her from winning the game trivially by choosing different content. Hence, such notions use the property that the scenarios are identical except for the senders and receivers (E_{SR}) to ensure that the messages are equal in both scenarios.

EXAMPLE: *An activist of the group is already well-known and communication with that person leads to persecution of Alice.*

Alice needs a protocol that hides whether a certain sender and receiver communicate with each other; cf. Section 4.2 motivation of the complex property M_{SR}. The resulting notion is *Sender-Receiver Pair Unlinkability ((SR)$\overline{L}$).*

EXAMPLE (CONT.): *Only few people participate in the protocol. Then, just using the protocol to receive (send) something, when the well known activist is acting as sender (receiver) threatens persecution.*

Alice needs a protocol that hides whether a certain sender and receiver actively participate at the same time or not; cf. Section 4.2 motivation of the complex property R_{SR}. The resulting notion is *Sender-Receiver Unobservability ((SR)$\overline{O}$).*

Sender-Receiver Linkability (Message Confidentiality). Senders and receivers can be learned in notions of this group, because they are not considered private. Hence, such notions include the property that the scenarios are identical, except for the messages (E_M) to ensure that the sender-receiver pairs are equal in both scenarios.

EXAMPLE: *Alice wants to announce her next demonstration. (1) Alice does not want the regime to learn the*

content of her message and block this event. (2) Further, she is afraid that the length of her messages could put her under suspicion, e.g. because activists tend to send messages of a characteristic length.

In (1) Alice needs a protocol that hides the content of the messages. However, the adversary is allowed to learn all other attributes, in particular the length of the message. Modeling this situation, the scenarios may differ solely in the message content; all other attributes must be identical in both scenarios, as they may not help the adversary distinguish between them. Beyond the above-described E_M, we must thus also request that the length of the messages $|M|$ is identical in both scenarios. The resulting notion is *Message Unobservability leaking Message Length ($M\overline{O} - |M|$)*[8].

In the second case (2), the protocol is required to hide the length of the message. The length of the messages thus may differ in the two scenarios, as the protocol will need to hide this attribute. Hence, we remove the restriction that the message length $|M|$ has to be equal in both scenarios from the above notion and end up with *Message Unobservability $M\overline{O}$.*

Both-Side Unobservability. Even the activity of a certain sender or receiver is hidden in notions of this group.

EXAMPLE (CONT.): *It is a risk for the activists, if the regime can distinguish between two leading activists exchanging the message "today" and two loyal regime supporters exchanging the message "tomorrow".*

In this case, Alice wants to disclose nothing about senders, receivers, messages or their combination. However, the adversary can learn the total number of communications happening in the ACN. Modeling this, we need to assure that for every communication in the first scenario, there exists one in the second. We achieve this by prohibiting the use of the empty communication with property $\emptyset$. This results in the notion *Unobservability ($\overline{O}$).*

EXAMPLE: *The regime knows that a demonstration is close, if the total number of communications transmitted over this protocol increases. It then prepares to block the upcoming event.*

To circumvent this, Alice needs a protocol that additionally hides the total number of communications. Modeling this, we need to allow the adversary to pick any two scenarios. Particularly, use of the empty communication $\lozenge$ is allowed. This is represented in the prop-

---

8 We stick to our naming scheme here, although we would commonly call this confidentiality.

erty that nothing needs to be equal in the two scenarios, $\aleph$, and results in the notion *Communication Unobservability (C$\overline{O}$)*. Note that this is the only notion where the existence of a communication is hidden. All other notions include $\emptyset$ and hence do not allow for the use of the empty communication.

6.1.2 Sender (and Receiver) Privacy Notions

These notions allow a greater freedom in picking the senders (or receivers: analogous notions are defined for receivers.).

Receiver-Message Linkability. The receiver-message relation can be disclosed in notions of this group. Hence, such notions include the property that the scenarios are identical except for the senders (E_S) to ensure the receiver-message relations are equal in both scenarios.

In *Sender-Message Unlinkability (SM$\overline{L}$)* the total number of communications and how often each user sends can be additionally learned. However, who sends which message is hidden. In *Sender-Frequency Unlinkability (SF$\overline{L}$)* the set of users and the total number of communications can be additionally disclosed. However, how often a certain user sends is hidden, since it can vary between the two scenarios. In *Sender Unobservability (S$\overline{O}$)*, the total number of communications can additionally be disclosed. However, especially the set of active senders U_b is hidden.

If a notion further includes the following abbreviations, the following information can be disclosed as well:

– *with User Number Leak* ($-|U|$): the number of senders that send something in the scenario
– *with Histogram Leak* ($-H$): the histogram of how many senders send how often
– *with Pseudonym Leak* ($-P$): which messages are sent from the same user

EXAMPLE: *Alice is only persecuted when the regime can link a message with compromising content to her* – she needs a protocol that at least provides $SM\overline{L}-P$. *However, since such a protocol does not hide the message content, the combination of all the messages she sent might lead to her identification.* Opting for a protocol that additionally hides the message combination (P), i.e. provides $SM\overline{L}$, can protect her from this threat. *Further, assuming most users send compromising content, and Alice's message volume is high, the regime might easily suspect her to be the origin of some compromising messages even if she is careful that the combination of her messages does not reidentify her* – she needs

a protocol that does not disclose her sending frequencies (Q) although the combination of her messages (P) might be learned, i.e. achieving $SF\overline{L}-P$. *However, Alice might fear disclosing the combination of her messages* - then she needs a protocol achieving at least $SF\overline{L}-H$, which hides the frequencies (Q) and the message combination (P), but discloses the sending histogram, i.e. how many people sent how many messages (H). *However, if multiple activist groups use the ACN actively at different time periods, disclosing the sending histogram H might identify how many activist groups exist and to which events they respond by more active communication* – to prevent this she needs a protocol that hides the frequencies Q and the histogram H, i.e. provides $SF\overline{L}$. *Further, not only sending a certain content, but also being an active sender (i.e. being in U) is prosecuted* she might want to pick a protocol with at least $S\overline{O}-P$. Again if she is afraid that leaking P or H together with the expected external knowledge of the regime would lead to her identification, she picks the corresponding stronger notion. *If the regime knows that senders in the ACN are activists and learns that the number of active senders is high, it blocks the ACN.* In this case at least $S\overline{O}$ should be picked to hide the number of senders ($|U|$).

EXAMPLE: *For the next protest, Alice sends two messages: (1) a location, and (2) a time. If the regime learns that both messages are from the same sender, they will block the place at this time even if they do not know who sent the messages.* Alice then needs a protocol that hides whether two communications have the same sender or not. We already explained how to model this with complex property T_S in Section 4.2. The resulting notion is Twice Sender Unlinkability($(2S)\overline{L}$).

Due to page limits the examples for the remaining notions can be found in Appendix C.

6.2 Analyzing Loopix's Privacy Goals

To check if we include currently-used privacy goals, we decide on a current ACN that has defined its goals based on an existing analytical framework and which has already been analyzed: the Loopix anonymity system [15]. In this section, we show that the privacy goals of Loopix map to notions we have defined (although the naming differs). Loopix aims for Sender-Receiver Third-Party Unlinkability, Sender online Unobservability and Receiver Unobservability.

Sender-Receiver Third-Party Unlinkability. Sender-Receiver Third-Party Unlinkability means that an adversary cannot distinguish scenarios where two receivers are switched:

"The senders and receivers should be unlinkable by any unauthorized party. Thus, we consider an adversary that wants to infer whether two users are communicating. We define *sender-receiver third party unlinkability* as the inability of the adversary to distinguish whether $\{S_1 \to R_1, S_2 \to R_2\}$ or $\{S_1 \to R_2, S_2 \to R_1\}$ for any concurrently online honest senders S_1, S_2 and honest receivers R_1, R_2 of the adversary's choice." [15]

The definition in Loopix allows the two scenarios to be distinguished by learning the first receiver. We interpret the notion such that it is only broken if the adversary learns a sender-receiver-pair, which we assume is what is meant in [15]. This means that the sender and receiver of a communication must be learned and is exactly the goal that motivated our introduction of complex properties: $(SR)\overline{L}$.

Unobservability. In sender online unobservability the adversary cannot distinguish whether an adversary-chosen sender communicates ($\{S \to\}$) or not ($\{S \not\to\}$):

"Whether or not senders are communicating should be hidden from an unauthorized third party. We define *sender online unobservability* as the inability of an adversary to decide whether a specific sender S is communicating with any receiver $\{S \to\}$ or not $\{S \not\to\}$, for any concurrently online honest sender S of the adversary's choice." [15]

Receiver unobservability is defined analogously.

Those definitions are open to interpretation. On the one hand, $\{S \not\to\}$ can mean that there is no corresponding communication in the other scenario. This corresponds to our $\Diamond$ and the definition of $LS\overline{O}$ and $LR\overline{O}$ in Appendix E.3. When a sender is not sending in one of the two scenarios, this means that there will be a receiver receiving in the other, but not in this scenario. Hence, $LS\overline{O}$ can be broken by learning about receivers and the two notions are equal. These notions are equivalent to $C\overline{O}$ (see Appendix E.3).

On the other hand, $\{S \not\to\}$ can mean that sender u does not send anything in this challenge. In this case, the receivers can experience the same behavior in both scenarios and the notions differ. We formulate this notion and argue its equivalence to $S\overline{O}$ (with a change in parameters) in Appendix E.4. This is equivalent to AnoA's sender anonymity α_{SA}. Analogously, Loopix's corresponding receiver notion is equivalent to $R\overline{O}$, which is even weaker than AnoA's receiver anonymity.

Remark. We do not claim that the Loopix system achieves or does not achieve any of these notions, since we based our analysis on the definitions of their goals, which were not sufficient to unambiguously derive the corresponding notions.

6.3 Relation to Existing Analysis Frameworks

In this section, we briefly introduce the existing frameworks based on indistinguishability games. We argue that our summary of notions includes all their notions[9] and therefore allows a comparison along this dimension. The resulting mapping is shown in Table 4 of Appendix D. Since the mapping of our properties to the notions of the other frameworks is obvious in most cases, we reason the remaining cases and concepts here and refer to the long version of this paper [13] for the complete verification.

AnoA Framework

AnoA [1] builds its privacy notions on (ϵ, δ) differential privacy and compares them to their interpretation of the terminology paper of Pfitzmann and Hansen [14].

Conceptually our model differs from AnoA's model in the definition of achieving a notion, batch queries, and the use of notions instead of anonymity functions. AnoA's definition of achieving a notion can be easily included (see Appendix B), if needed. In AnoA, the adversary gets information after every communication. This is equivalent to multiple batches of size one in our case.

AnoA's challenger does not only check properties, but modifies the batches with the *anonymity functions*. However, the modification results in one of at most four batches. We require those four batches (as combination of scenario and instances) as input from the adversary, because it is more intuitive that all possible scenarios stem from the adversary. This neither increases nor reduces the information the adversary learns, since she knows the challenger algorithm.

Bohli's Framework

Bohli and Pashalidis [3] build a hierarchy of application-independent privacy notions based on what they define as "interesting properties", that the adversary is or is not allowed to learn. Additionally, they compare their notions to Hevia's, which we introduce next, and find equivalences.

To achieve the mapping, we need to interpret one property of Bohli's framework for ACNs. Our message partitionings (P, P') group the messages by their sender/receiver. However, Bohli's corresponding linking

---

9 Where necessary, we have interpreted them for ACNs and broken them down to the general observable information.

relation groups the indexes of the outputs of the analyzed system. Since messages are usually the interesting output elements, the adversary tries to link in ACNs; we consider this as a suitable mapping when analyzing ACNs.

Hevia's Framework

Hevia and Micciancio [12] define scenarios based on message matrices. Those message matrices specify who sends what message to whom. Notions restrict different communication properties like the number or set of sent/received messages per fixed user, or the number of total messages. Further, they construct a hierarchy of their notions and give optimal ACN protocol transformations that, when applied, lead from weaker to stronger notions.

In contrast, our model considers the order of communications. Analyzing protocol models that ignore the order will lead to identical results. However, protocol models that consider the order do not achieve a notion – although they would in Hevia's framework, if an attack based on the order exists.

Most of Hevia's notions are already shown to match Bohli's with only one batch [3]. However, we have to correct two mappings: in [3] Hevia's strong sender anonymity (SA^*), which requires the number of messages a receiver receives to be the same in both scenarios was mistakenly matched to Bohli's sender weak unlinkability (S/WU^+), in which every sender sends the same number of messages in both scenarios. Hence, the sender and receiver restrictions become confused and it needs to be mapped to Bohli's receiver weak unlinkability (R/WU^+) instead. The same reasoning leads to Bohli's sender weak unlinkability (S/WU^+) as the mapping for Hevia's strong receiver anonymity (RA^*).

Gelernter's Framework

Gelernter and Herzberg [10] extend Hevia's framework to include corrupted participants. Additionally, they show that under this strong adversary an ACN protocol achieving the strongest notions exists. However, they prove that any ACN protocol with this strength has to be inefficient, i.e. the message overhead is at least linear in the number of honest senders. Further, they introduce relaxed privacy notions that can be efficiently achieved.

The notions of Gelernter's framework build on Hevia's and add corruption, which we do not discuss in this work, but include in the long version of this paper [13]. However, the relaxed notions are not solely an extension regarding corruption. In Appendix E.2 we formalize them and shown to be equivalent to two previously defined notions.

7 Hierarchy

Next, we want to compare all notions and establish their hierarchy. To do this, for any pair of notions we analyze which one is stronger than, i.e. implies, the other. This means, any ACN achieving the stronger notion also achieves the weaker (implied) one. Our result is shown in Figure 3, where all arrow types represent implications, and is proven as Theorem 1 below. Further, obvious implications between every notion $S\overline{O}\{X\}$, $R\overline{O}\{X\}$ and X exist, since $S\overline{O}\{X\}$ only adds more possibilities to distinguish the scenarios. However, to avoid clutter we do not show them in Figure 3.

Theorem 1. *The implications shown in Figure 3 hold.*

Proof sketch. We prove every implication $X_1 \Rightarrow X_2$ by an indirect proof of the following outline: Given an attack on X_2, we can construct an attack on X_1 with the same success. Assume a protocol has X_1, but not X_2. Because it does not achieve X_2, there exists a successful attack on X_2. However, this implies that there exists a successful attack on X_1 (we even know how to construct it). This contradicts that the protocol has X_1.[10] Due to this construction of the proof, the implications are transitive.

We use different arrow styles in Figure 3 to partition the implications into those with analogous proofs.

→ follow from the definition of the notions.

⋯▸ hold, because of the following and analogous arguments: every attack against $S\overline{O}$ is valid against $S\overline{O}\{RM\overline{L} - P'\}$: Because of E_S the receiver-message pairs of the communications input to the protocol are the same in both scenarios. Hence, every receiver receives the same messages, i.e. Q' and P' are fulfilled.

⇢ $X_1 \Rightarrow X_2$ hold, because of the following and analogous arguments:[11] given attack $\mathcal{A}_2$ on $(SR)\overline{O}$. We construct two attacks $\mathcal{A}_1'$ and $\mathcal{A}_1''$ against $R\overline{O}$ and show that one of those has at least the desired success.

---

10 In AnoA, Bohli's and Hevia's framework some of these implications are proved for their notions in the same way.

11 For $S\overline{O} \Rightarrow (SR)\overline{O}$ (or $S\overline{O} \Rightarrow (SM)\overline{O}$) pick challenge rows differently; for $b = 0 : a = a'$ and for $b = 1 : a = 1 - a'$ to ensure that receivers (or messages) are equal.
For $SM\overline{L} \Rightarrow (SM)\overline{L}$, (or $RM\overline{L} \Rightarrow (RM)\overline{L}$, $SM\overline{L} - P \Rightarrow (SR)\overline{L}$) replace the challenge row with the corresponding two rows.
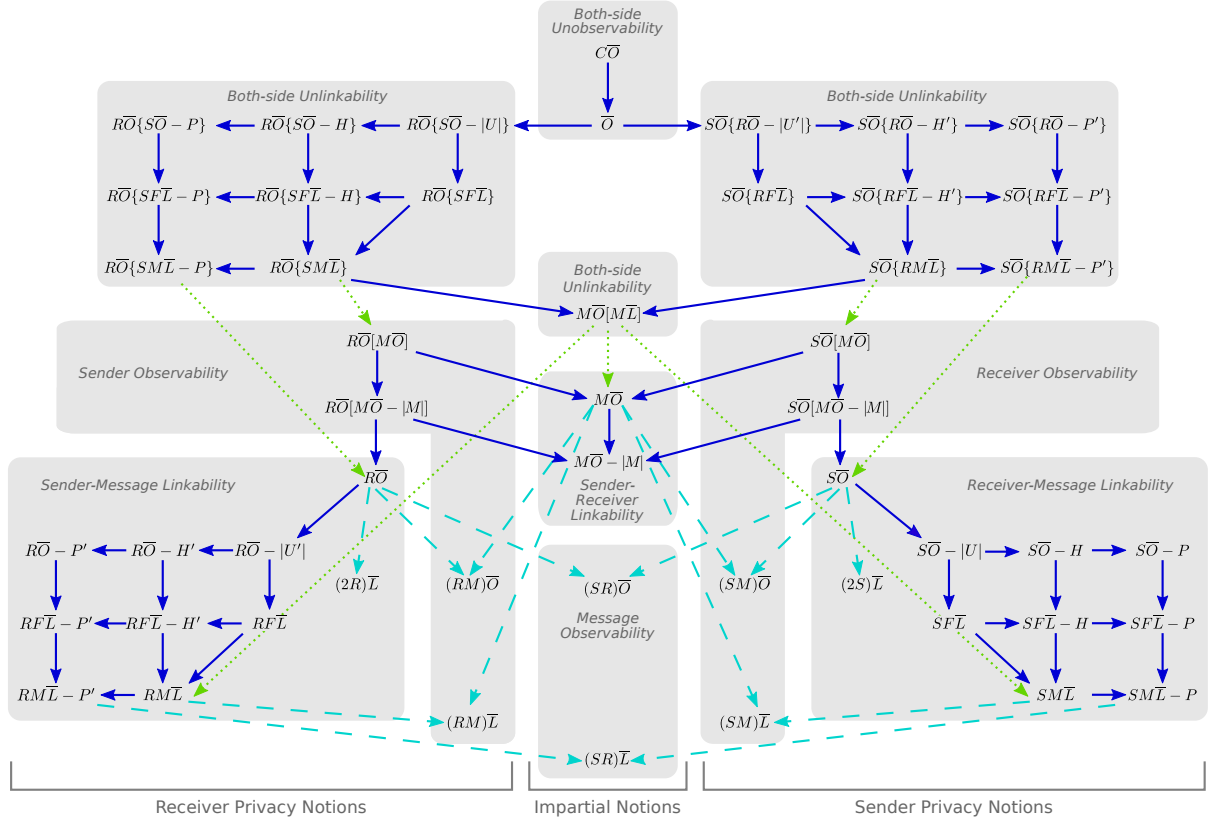
Fig. 3. Our hierarchy of privacy notions divided into sender, receiver and impartial notions and clustered by leakage type. Table 2 provides definitions for the presented notions based on properties. Table 5 gives an overview on all properties. For a summary of the naming scheme, see Table 3 of Appendix D.

We construct attacks $\mathcal{A}'_1$ and $\mathcal{A}''_1$ by picking $a' = 0$ and $a'' = 1$. These shall replace a, which would be picked randomly by the challenger in $(SR)\overline{O}$ to determine the instance. In $\mathcal{A}'_1$ we use the communications of $\mathcal{A}_2$ corresponding to $a' = 0$ (for $b = 0$ and $b = 1$) as the challenge row, whenever a batch in $\mathcal{A}_2$ includes a challenge row. In $\mathcal{A}''_1$ we analogously use the communications corresponding to $a'' = 1$.

$\mathcal{A}'_1$ and $\mathcal{A}''_1$ are valid against $R\overline{O}$: Because of the fixed $a = a'$ or $a = a''$, the senders of challenge rows are the same in both scenarios. Since messages are also equal in $(SR)\overline{O}$, the sender-message pairs are fixed (E_R). Since $\mathcal{A}_2$ is an successful attack on $(SR)\overline{O}$ and $\mathcal{A}'_1$ and $\mathcal{A}''_1$ against $R\overline{O}$ only fix the otherwise randomly-picked a, one of the two newly-constructed attacks successfully breaks $R\overline{O}$. For the case of multiple challenges we refer the reader to the extended version. □

Further, our hierarchy is complete in the sense that no implications are missing:

Theorem 2. *For all notions X_1 and X_2 of our hierarchy, where $X_1 \implies X_2$ is not proven or implied by transitivity, there exists an ACN protocol achieving X_1, but not X_2.*

Proof sketch. We construct the protocol in the following way: Given a protocol Π that achieves X'_1 (X_1 itself or a notion that implies X_1), let protocol Π' run Π and additionally output some information I. We argue that learning I does not lead to any advantage in distinguishing the scenarios for X_1. Hence, Π' achieves X_1. We give an attack against X_2 where learning I allows the scenarios to be distinguished. Hence, Π' does not achieve X_2. Further, we use the knowledge that $\implies$ is transitive[12].

Some concrete cases are shown in Appendix E.1. We provide the complete list of proofs in the long version of this paper [13]. □

---

12 If $X_1 \implies X_2$ and $X_1 \not\implies X_3$, it follows that $X_2 \not\implies X_3$.

8 Discussion

In this section, we present the lessons learned while creating our framework.

Learning about privacy goals. The need for formal definitions is emphasized by the mapping of Loopix's privacy goals to notions as example that less formal alternatives leave room for interpretation. Further, a result like our hierarchy would be much harder to achieve without formal definitions.

These definitions allow us to point out the relation of privacy and confidentiality ($M\overline{O} - |M|$). The way we ordered the notions in the hierarchy allows easy identification the notions implying $M\overline{O} - |M|$ (the middle of the upper part). Note that any privacy notion implying $M\overline{O} - |M|$ can be broken by distinguishing the message's content. Further, nearly all those notions also imply $M\overline{O}$ and hence, all such notions can be broken by learning the message length.

Our formal definitions also enabled the comparison of existing frameworks. Excluding differences in the adversarial model, quantifications and restrictions that do not apply to all ACNs, we observe that equivalent definitions are often defined independently by the authors of the analytical frameworks. For this reason, we included the notions of the other frameworks in our hierarchy in Figure 4 of Appendix E.4. $\overline{O}$, $S\overline{O} - P$, $SM\overline{L} - P$, $R\overline{O}\{SM\overline{L}\}$ and $SM\overline{L}$ are defined (under different names) in multiple works; $S\overline{O}$ is even defined in all works.

Although previous work includes equivalent definitions, we realized that some notions are still missing. For example, we added weak notions like $(SM)\overline{L}$, $(RM)\overline{L}$ and $(SR)\overline{L}$ because they match our understanding of anonymity. Our understanding was confirmed by the analysis of Loopix' goals. Further, we defined all analogous notions for all communication parties involved (senders and receivers) as real-world application define which party is more vulnerable. For the concrete applications we refer the reader to Section 6.1.

Consequently, we present a broad selection of privacy notions. We are aware that understanding them all in detail might be a challenging task, so we want to provide some intuitions and preferences, based on what we know and conjecture. We expect the lower part of the hierarchy to be more important for ACNs as [10] already includes an inefficiency result for $S\overline{O}$ and thus for all notions implying $S\overline{O}$. As a first guess, we think $S\overline{O}$, if higher overhead is manageable, $SF\overline{L}$, $SM\overline{L}$, $(SM)\overline{L}$ (and receiver counterparts), $M\overline{O} - |M|$ and $(SR)\overline{L}$ are

the most popular notions for ACNs. Further, we want to add some results concerning two well-known systems to ease intuition. [1]'s analysis of Tor results in a small, but non-negligible probability to break $S\overline{O}$ and thus Tor does not achieve $S\overline{O}$ with our strict definition. Classical DC-Nets, on the other hand, do achieve at least $S\overline{O} - P$ [10].

Correcting Inconsistencies. While the above similarities most likely stem from the influence of prior informal work on privacy goals, attempts to provide concrete mappings have led to contradictions. The AnoA framework maps its notions to their interpretation of Pfitzmann and Hansen's terminology. Pfitzmann and Hansen match their terminology to the notions of Hevia's framework. This means that, notions of AnoA and Hevia's framework are indirectly mapped. However, those notions are not equivalent. While AnoA's sender anonymity and Hevia's sender unlinkability are both mapped to Pfitzmann and Hansen's sender anonymity, they differ: In Hevia's sender unlinkability the number of times every sender sends can leak to the adversary, but in AnoA's sender anonymity it cannot.

We belive that AnoA's sender anonymity should be called sender unobservability, which is also our name for the corresponding notion. This follows the naming proposal of Pfitzmann and Hansen and their mapping to Hevia. It is also more suitable because AnoA's sender anonymity can be broken by learning whether a certain sender is active, i.e. sends a real message, in the system ($u \in U_b$). In order to achieve this notion, all senders have to be unobservable. To verify this, we looked at how the notions of AnoA have been used. For example in [5] the protocol model contains an environment that lets all senders send randomly. Hence, U_b is hidden by the use of this environment. We consider that the information that is allowed to be disclosed should instead be part of the notion and not modified by an environment. Only then are the notions able to represent what information is protected by the protocol.

Another lesson learned by comparing privacy notions is the power of names, because they introduce intuitions. The fact that Hevia's strong sender anonymity is equivalent to Bohli's receiver weak unlinkability seems counter-intuitive, since a sender notion is translated to a receiver notion. This might also be the reason for the incorrect mapping in [3]. However, Bohli's receiver weak unlinkability is named this way because receivers are the "interesting" users, whose communication is restricted. It does not restrict senders in any way and hence should be, in most cases, easier to break according to some information about the sender. This is why

we and Hevia have classified it as a sender notion. An analogous argument explains why Bohli's receiver weak anonymity R/WA implies the restricted case of Bohli's sender strong anonymity $S/SA°$.

Long Version [13]. Besides giving more technical details, we focus on making our results easier to apply for practitioners in the long version of thus paper by presenting an analysis framework, along with a how-to-use section. The extended version includes different parts of the adversary model, like user corruption and limiting the number of adversarial users, and discusses how typical attacks, like n-1, intersection and active attacks (e.g. delaying or dropping messages), apply to our framework. To further simplify the proofs practitioners have to make, it allows privacy goals to be quantified by using multiple challenges or multiple challenge rows and includes results on how the limited case of challenge rows generalizes to more, such that only the limited case needs to be proven. Further, as we are aware that our strict definition of achieving a notion might not work for some practical cases, we point out the relaxed definition that allows for a non-negligible distinguishing probability. However, none of those extensions limits or contradicts the results regarding the hierarchy of privacy notions built from observable properties that we presented here, as they work independently.

9 Conclusion and Future Work

We expressed privacy goals formally as privacy notions. We first presented their basic building blocks: properties. Those properties cover the observable information of communications, which is either required to remain private or allowed to be learned by an adversary, depending on the goal. We formally specified privacy goals from ACNs and sorted them into a proven hierarchy, according to their strength. This means, for every pair of notions, we know which one is the stronger; or if they do not imply each other. As a result, we resolved inconsistencies between existing analytical frameworks and built the foundations to understand the strengths and weaknesses of ACNs better, which helps analyzing and building improved ACNs.

Future Work. As we mentioned in the discussion, providing more intuitions and understanding the significance of notions is necessary. Therefore, analogous to the analysis of Loopix's privacy goals, more current ACNs can be analyzed to understand which parts of the hierarchy they cover. This can also identify gaps in re-

search; privacy goals for which ACNs are currently missing. Further, a survey of goals in greater depth would be useful to identify the most important notions in the hierarchy and to provide intuitions and thus ease deciding on the correct notions for practitioners.

Additionally, such a survey helps to understand the relationships between currently-employed privacy enhancing technologies. Finally, this understanding and the knowledge about how notions are related and differ can be used to define general techniques that strengthen ACNs.

Beyond that, an investigation of the applicability of our notions and hierarchy to other areas, like e.g. anonymous payment channels, would be interesting.

Acknowledgements

References

[1] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi. AnoA: A framework for analyzing anonymous communication protocols. *Journal of Privacy and Confidentiality*, 2017.

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

[3] J.-M. Bohli and A. Pashalidis. Relations among privacy notions. *TISSEC*, 2011.

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

[5] D. Chaum, F. Javani, A. Kate, A. Krasnova, J. de Ruiter, A. T. Sherman, and D. Das. cMix: Anonymization by high-performance scalable mixing. *USENIX Security*, 2016.

[6] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 1981.

[7] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing privacy enhancing technologies*, 2001.

[8] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, Naval

Research Lab Washington DC, 2004.

[9] C. Dwork, A. Roth, and others. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 2014.

[10] N. Gelernter and A. Herzberg. On the limits of provable anonymity. In *WPES*, 2013.

[11] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of computer and system sciences*, 1984.

[12] A. Hevia and D. Micciancio. An indistinguishability-based characterization of anonymous channels. *Lecture Notes in Computer Science*, 2008.

[13] C. Kuhn, M. Beck, S. Schiffner, E. Jorswieck, and T. Strufe. On privacy notions in anonymous communication. In *arXiv preprint arXiv:1812.05638*, 2018.

[14] A. Pfitzmann and M. Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. 2010.

[15] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, and G. Danezis. The loopix anonymity system. In *26th USENIX Security Symposium, USENIX Security*, 2017.

[16] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *TISSEC*, 1998.

[17] B. Zantout and R. Haraty. I2P data communication system. In *ICN*, 2011.

A Challenger

This section describes the queries to the challenger $Ch(\Pi, X, b)$.

Batch Query. The batches $\underline{r_0}, \underline{r_1}$ that the adversary chooses for the two scenarios are represented in batch queries. When the challenger receives a batch query, it will validate the communications that would be input to Π for $b = 0$ and $b = 1$ as explained below. If all criteria are met so far, it checks that the properties of the privacy notion X are met by using stored information about the past batches and the instances for both scenarios $\underline{r_0}^a, \underline{r}^a_1, a \in \{0, 1\}$. Finally, it runs the instance belonging to the challenge bit b of this game and the for this challenge randomly chosen instance bit a, if the properties are matched. Otherwise, it returns $\perp$ and aborts the experiment. Running the scenario in the ACN protocol will return information that is forwarded to the adversary. This information is what an adversary is assumed to be able to observe.

Protocol Query. Protocol queries allow the adversary e.g. to compromise parts of the network, set parameters of the ACN protocol or use other functionalities modeled in the protocol model, like e.g. active attacks. The meaning and validity of those queries is specific to the analyzed ACN protocol.

Switch Stage Query. If this query occurs and it is allowed, i.e. the notion contains a relevant property, the stage is changed from 1 to 2.

Validate Communications. If the analyzed ACN protocol specifies restrictions of senders and receiver-message pairs, their validity is checked by this function.

Remark to simple properties and instances. In case the notion only uses simple properties, the challenger will pick $a = 0$ and check the properties for $\underline{r_1}_j = r^0_{1_j}$ and $\underline{r_0}_j = r^0_{0_j}$. In case the notion uses a combination of simple and complex properties, the challenger will check the simple properties for any pair $\underline{r_1}_j = \underline{r}^a_{1_j}$ and $\underline{r_0}_j = \underline{r}^{a'}_{0_j}$ resulting by any $a, a' \in \{0, 1\}$.

B Achieving (ϵ, δ)-X

For some use cases, e.g. if the court of your jurisdiction requires that the sender of a critical content can be identified with a minimal probability of a certain threshold e.g. 70%, a non-negligible δ is suitable. Hence, we allow to specify the parameter of δ and include the well-known concept of differential privacy [9] as AnoA does in the following Definition:

Definition 11 (Achieving $(\epsilon, \delta) - X$). *An ACN protocol Π is (ϵ, δ) -X with $\epsilon \geq 0$ and $0 \leq \delta \leq 1$, iff for all PPT algorithms $\mathcal{A}$:*

$$Pr[0 = \langle \mathcal{A} \mid Ch(\Pi, X, 0) \rangle] \leq$$
$$e^\epsilon Pr[0 = \langle \mathcal{A} \mid Ch(\Pi, X, 1) \rangle] + \delta.$$

Note that ϵ describes how close the probabilities of guessing right and wrong have to be. This can be interpreted as the quality of privacy for this notion. While δ describes the probability with which the ϵ-quality can be violated. Hence, every ACN protocol will achieve $(0, 1) - X$ for any notion X, but this result does not guarantee anything, since with probability $\delta = 1$ the ϵ-quality is not met.

Note Π is $(0, \delta) - X$ for a negligible δ is equivalent to the first definition of Π achieves X.

C Remaining Examples

Impartial Notions: Both-Side Message Unlinkability. Notions of this group are broken if the sender-message or receiver-message relation is revealed.

EXAMPLE: *The activists know that their sending and receiving frequencies are similar to regime supporters'*

and that using an ACN is in general not forbidden, but nothing else. Even if the content and length of the message ($M\overline{O}$) and the sender-receiver relationship (($SR)\overline{L}$) is hidden, the regime might be able to distinguish uncritical from critical communications, e.g. whether two activists communicate "Today" or innocent users an innocent message. In this case, the regime might learn that currently many critical communications take place and improves its measures against the activists.

In this case, the activists want a protocol that hides the communications, i.e. relations of sender, message and receiver. However, as using the protocol is not forbidden and their sending frequencies are ordinary, the adversary can learn which users are active senders or receivers and how often they sent and receive. Modeling this, the users need to have the same sending and receiving frequencies in both scenarios Q, Q', since it can be learned. However, everything else needs to be protected and hence, can be chosen by the adversary. This corresponds to the notion *Message Unobservability with Message Unlinkability ($M\overline{O}[M\overline{L}]$).*

Sender Privacy Notions: Receiver Observability. In notions of this group the receiver of each communication can be learned. Hence, such notions include the property that the scenarios are equal except for the senders and messages (E_{SM}) to ensure that they are equal in both scenarios.

EXAMPLE: *Consider not only sending real messages is persecuted, but also the message content or any combination of senders and message contents is exploited by the regime. If the regime e.g. can distinguish activist Alice sending "today" from regime supporter Charlie sending "see u", it might have learned an information the activists would rather keep from the regime. Further, either (1) the activists know that many messages of a certain length are sent or (2) they are not sure that many messages of a certain length are sent.*

In case (1), Alice needs a ACN, that hides the sender activity, the message content and their combination. However, the adversary can especially learn the message length. Modeling this, beyond the above described E_{SM}, the message lengths have to be equal $|M|$. This results in the notion *Sender Unobservability with Message Unobservability leaking Message Length ($S\overline{O}[M\overline{O} - |M|]$).* Note that in $S\overline{O}[M\overline{O}-|M|]$ the properties of $M\overline{O}-|M|$ are included and further the senders are allowed to differ in the two scenarios. The second case (2) requires a protocol that additionally hides the message length. Hence, in modeling it we remove the property that the message lengths are equal $|M|$ from the above notion.

This results in *Sender Unobservability with Message Unobservability ($S\overline{O}[M\overline{O}]$).*

EXAMPLE: *Alice's demonstration is only at risk if the regime can link a message with a certain content to her as a sender with a non negligible probability. Then at least Sender-Message Pair Unlinkability (($SM)\overline{L}$), which is defined analogous to ($SR)\overline{L}$ is needed.*

EXAMPLE (CONT.): *However, ($SM)\overline{L}$ only allows Alice to claim that not she, but Charlie sent a critical message m_a and the regime cannot know or guess better. Now assume that Dave is also communicating, then the regime might be able to distinguish Alice sending m_a, Charlie m_c and Dave m_d from Alice sending m_d, Charlie m_a and Dave m_c. In this case, it might not even matter that Alice can claim that Charlie possibly sent her message. The fact that when comparing all three communications that possibly happened, Alice is more likely to have sent the critical message m_a means a risk for her.*

To circumvent this problem Alice needs a protocol that not only hides the difference between single pairs of users, but any number of users. Modeling this, instead of the complex property M_{SM}, we need to restrict that the active senders' sending frequencies are equal, i.e. $SM\overline{L}$.

EXAMPLE: *In another situation our activists already are prosecuted for being a sender while a message with critical content is sent.*

In this case at least *Sender-Message Pair Unobservability (($SM)\overline{O}$), which is defined analogous to ($SR)\overline{O}$* is needed.

Analogous notions are defined for receivers.

Sender Privacy Notions: Both-Side Message Unlinkability. As explained with the example before in the case that Alice does not want any information about senders, receivers and messages or their combination to leak, she would use $\overline{O}$. However, the privacy in this example can be tuned down, if she assumes that the regime does not have certain external knowledge or that the users are accordingly careful. As explained for the Sender Notions with Receiver-Message Linkability before, in this case we might decide to allow $U', |U'|, Q', H', P'$ to leak.

If a notion $X \in \{R\overline{O}, R\overline{O} - |U'|, R\overline{O} - H', R\overline{O} - P', RF\overline{L}, RF\overline{L} - H', RF\overline{L} - P', RM\overline{L}, RM\overline{L} - P'\}$ is extended to *Sender Unobservability by X ($S\overline{O}\{X\}$),* the leaking of the sender-message relation is removed. This is done by removing E_R. Since the attacker now has a greater degree of freedom in choosing the senders and is (if at all) only restricted in how she chooses the receivers and messages, this is a special strong kind of

Sender Unobservability. Analogous notions are defined for receivers.[13]

D Additional Tables and Lists

Usage	Explanation
$D \in \{S, R, M\}$	Dimension $\in$ {Sender, Receiver, Message}
Dimension D not mentioned	Dimension can leak
Dimension D mentioned	Protection focused on this dimension exists
$D\overline{O}$	not even the active participating items regarding D leak,(e.g. $S\overline{O}$: not even U leaks)
$DF\overline{L}$	active participating items regarding D can leak, but not which exists how often (e.g. $SF\overline{L}$: Uleaks, but not Q)
$DM\overline{L}$	active participating items regarding D and how often they exist can leak (e.g. $SM\overline{L}$: U, Q leaks)
$X - Prop$, $Prop \in \{|U|, H, P, |U'|, H', P', |M|\}$	like X but additionally Prop can leak
$(D_1 D_2)\overline{O}$	uses $R_{D_1 D_2}$; active participating items regarding D_1, D_2 do not leak, (e.g. $(SR)\overline{O}$: R_{SR})
$(D_1 D_2)\overline{L}$	uses $M_{D_1 D_2}$; active participating items regarding D_1, D_2 can leak, (e.g. $(SR)\overline{L}$: M_{SR})
$(2D)\overline{L}$	uses T_D; one active participating item regarding D has to be identified twice, (e.g. $(2S)\overline{L}$: T_S)
$\overline{O}$	short for $S\overline{O}R\overline{O}M\overline{O}$
$M\overline{O}[M\overline{L}]$	short for $M\overline{O}(SM\overline{L}, RM\overline{L})$
$S\overline{O}\{X\}$	short for $S\overline{O}M\overline{O}X$
$D_1 X_1[D_2 X_2]$	D_1 is dominating dimension, usually D_1 has more freedom, i.e. X_2 is a weaker restriction than X_1
$C\overline{O}$	nothing can leak (not even the existence of any communication)

Table 3. Naming Scheme

E Proof Sketches

Here we include the proof sketches mentioned before.

E.1 For Implication Completeness

Proof Sketch (continued). Tables 8 gives the idea of some proofs. $|U'|$ means the number of receivers is leaked. The other abbreviations are used analogously. The attack is shortened to the format ⟨(communications of instance 0 scenario 0),(communications of instance 1 of scenario 0)⟩,⟨(communications of instance 0 scenario 1),(communications of instance 1 of scenario 1)⟩ (if both instances of the scenario are equal, we shorten to:⟨(communications of instance 0 scenario 0)⟩,⟨(communications of instance 0 scenario 1)⟩) and all not mentioned elements are equal in both scenarios. m_0, m_1, m_2, m_3 are messages with $|m_0| < |m_1|$, $|m_2| = |m_3|$ and $m_0 \neq m_1 \neq m_2 \neq m_3$; u_0, u_1, u_2 senders and u'_0, u'_1, u'_2 receivers. □

Framework	Notion	Equivalent to
AnoA	α_{SA}	$S\overline{O}$
	α_{RA}	$R\overline{O}[M\overline{O} - |M|]$
	α_{REL}	$(SR)\overline{O}$
	α_{UL}	$(2S)\overline{L}$
Bohli's	$S/SA = R/SA$	$\overline{O}$
	R/SUP	$S\overline{O}\{R\overline{O} - |U'|\}$
	R/WUP	$S\overline{O}\{R\overline{O} - H'\}$
	R/PS	$S\overline{O}\{R\overline{O} - P'\}$
	R/SUU	$S\overline{O}\{RF\overline{L}\}$
	R/WUU	$S\overline{O}\{RF\overline{L} - H'\}$
	R/AN	$S\overline{O}\{RF\overline{L} - P'\}$
	R/WU	$S\overline{O}\{RM\overline{L}\}$
	R/WA	$S\overline{O}\{RM\overline{L} - P'\}$
	$S/SA°$	$S\overline{O}$
	$S/SUP°$	$S\overline{O} - |U|$
	$S/WUP°$	$S\overline{O} - H$
	$S/PS°$	$S\overline{O} - P$
	$S/SUU°$	$SF\overline{L}$
	$S/WUU°$	$SF\overline{L} - H$
	$S/AN°$	$SF\overline{L} - P$
	$S/WU°$	$SM\overline{L}$
	$S/WA°$	$SM\overline{L} - P$
	$S/X, R/X°$	analogous
Hevia's	UO	$C\overline{O}$
	SRA	$\overline{O}$
	SA^*	$S\overline{O}\{RM\overline{L}\}$
	SA	$S\overline{O}$
	UL	$M\overline{O}[M\overline{L}]$
	SUL	$SM\overline{L}$
	RA^*, RUL, RA	analogous
Gelernter's	$R_{SA}^{H,\tau}$	$R_{SA}^{H,\tau} \iff S\overline{O} - P$
	$R_{SUL}^{H,\tau}$	$R_{S\overline{L}}^{H,\tau} \iff SM\overline{L} - P$
	R_X	analogous Hevia: $\langle X \rangle$

Table 4. Equivalences, $\langle X \rangle$ equivalence of X used

E.2 For Notions of other Frameworks

We define new notions as $R_{SA}^{H,\tau} = \aleph \land G$ and $R_{S\overline{L}}^{H,\tau} = \aleph \land Q \land G$ that are equivalent to some of the already introduced notions to make the mapping to the Gelernter's notions obvious. They use a new property G, in which scenarios are only allowed to differ in the sender names.

Definition 12 (Property G). *Let $\mathcal{U}$ be the set of all possible senders, L_{b_i} the sender-message linking for scenario $b \in \{0,1\}$. We say that G is met, iff a permutation perm on $\mathcal{U}$ exists such that for all $(u, M) \in L_{0_k}$: $(perm(u), M) \in L_{1_k}$.*

Theorem 3. *It holds that*

$$R_{SA}^{H,\tau} \iff S\overline{O} - P,$$
$$R_{S\overline{L}}^{H,\tau} \iff SM\overline{L} - P.$$

Proof sketch. Analogous to Theorem 1: See long version for details. □

---

13 Note that $S\overline{O}\{R\overline{O}\} = R\overline{O}\{S\overline{O}\} = \overline{O}$.
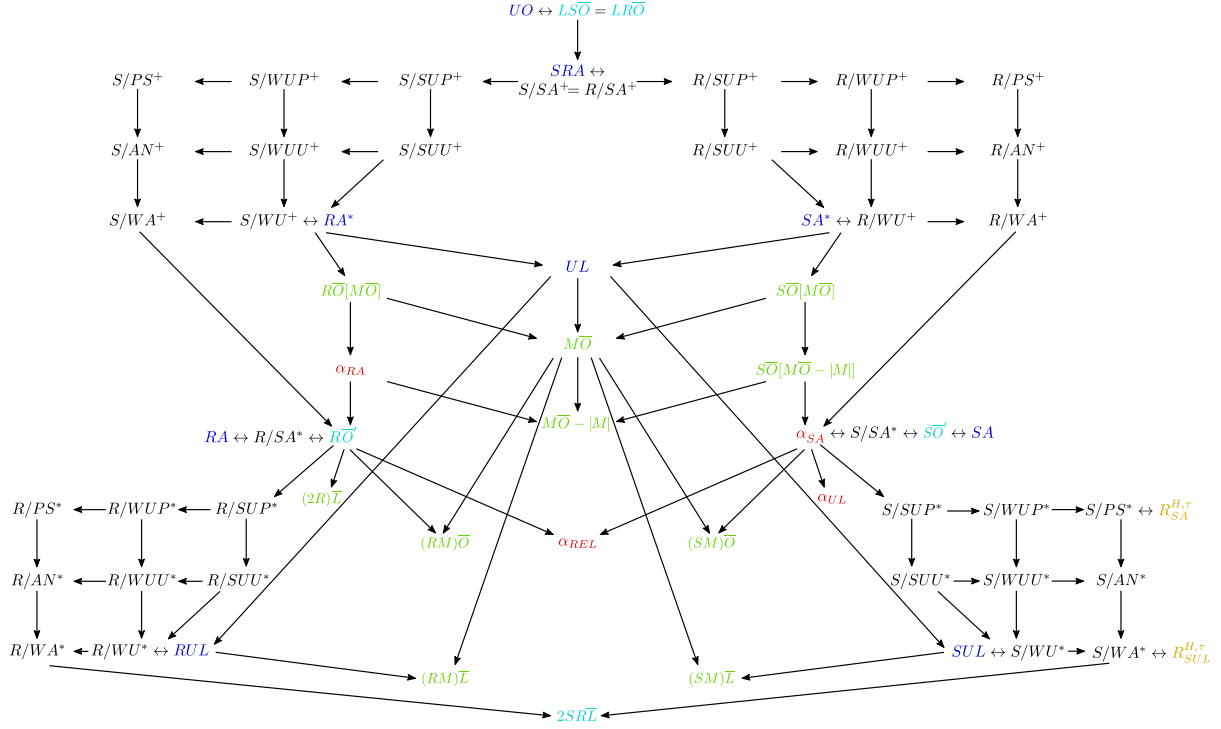
Fig. 4. Our hierarchy with the mapping of the other works (Bohli's, AnoA, Hevia's, Gelernter's framework, Loopix's ACN and new notions)

E.3 For Loopix's Notions 1

We define $LS\overline{O}$ and $LR\overline{O}$ according to Table 9. Therefore, we need the property that if something is sent in both scenarios, it is the same.

Definition 13 ($E_\Diamond$). *Let the checked batches be $\underline{r_0}, \underline{r_1}$, including communications $r_{0j}, r_{1j}, j \in \{1, \dots l\}$. We say $E_\Diamond$ is met, iff for all $j \in \{1, \dots l\}$:*

$$E_\Diamond : r_{0j} = r_{1j} \lor r_{0j} = \Diamond \lor r_{1j} = \Diamond$$

Theorem 4. *It holds that $C\overline{O} \iff LS\overline{O}$.*

Proof sketch. Implications are proven analogously to the ones in Theorem 1. $C\overline{O} \implies LS\overline{O}$ by definition. $LS\overline{O} \implies C\overline{O}$ because for every challenge row (r_0, r_1) in the attack on $C\overline{O}$, we can create two batches $(r_0, \Diamond)$ and $(\Diamond, r_1)$. □

E.4 For Loopix's Notions 2

We define $S\overline{O}'$ and $R\overline{O}'$ according to Table 9. To formulate these notions we need a new property that some sender/receiver is not participating in any communication in the second scenario:

Definition 14 (Property $\nrightarrow$). *Let u be the sender of the first scenario in the first challenge row of this chal-*

lenge. We say that $\nrightarrow$ is fulfilled iff for all $j : u_{1j} \neq u$. (Property $\nrightarrow'$ is defined analogously for receivers.)

Theorem 5. *It holds that $S\overline{O} \iff S\overline{O}'$.*

Proof sketch. Analogously to Theorem 1. See long version for details. □

Symbol	Description
U/U'	Who sends/receives is equal for both scenarios.
Q/Q'	Which sender/receiver sends/receives how often is equal for both scenarios.
H/H'	How many senders/receivers send/receive how often is equal for both scenarios.
P/P'	Which messages are sent/received from the same sender/receiver is equal for both scenarios.
$\|U\|/\|U'\|$	How many senders/receivers communicate is equal for both scenarios.
$\|M\|$	Messages in the two scenarios always have the same length.
E_S	Everything but the senders is identical in both scenarios.
E_R, E_M	analogous
E_{SM}	Everything but the senders and messages is identical in both scenarios.
E_{RM}, E_{SR}	analogous
$\aleph$	nothing will be checked; always true
$E_\diamond$	If something is sent in both scenarios, the communication is the same.
$\emptyset$	In every communication something must be sent.
R_{SR}	Adversary picks two sender-receiver-pairs. One of the senders and one of the receivers is chosen randomly. For b=0 one of the adversary chosen sender-receiver pairs is drawn. For b=1 the sender is paired with the receiver of the other pair.
R_{SM}, R_{RM}	analogous
T_S	Adversary picks two senders. The other sender might send the second time (stage 2). For b=0 the same sender sends in both stages, for b=1 each sender sends in one of the stages.
T_R	analogous
M_{SR}	Adversary picks two sender-receiver-pairs. Sender-receiver-pairs might be mixed. For b=0 both adversary chosen sender-receiver-pairs communicate. For b=1 both mixed sender-receiver-pairs communicate.
M_{SM}, M_{RM}	analogous

Table 5. Properties

Symbol	Description
$\overline{SO}\{\overline{RO} - \|U'\|\}$	Sender/Message Unobservability with Receiver Unobservability leaking User Number
$\overline{SO}\{\overline{RO} - H'\}$	Sender/Message Unobservability with Receiver Unobservability leaking Histogram
$\overline{SO}\{\overline{RO} - P'\}$	Sender/Message Unobservability with Receiver Unobservability leaking Pseudonym
$\overline{SO}\{R F \overline{L}\}$	Sender/Message Unobservability with Receiver-Frequency Unlinkability
$\overline{SO}\{R F \overline{L} - H'\}$	Sender/Message Unobservability with Receiver-Frequency Unlinkability leaking Histogram
$\overline{SO}\{R F \overline{L} - P'\}$	Sender/Message Unobservability with Receiver-Frequency Unlinkability leaking Pseudonym
$\overline{SO}\{R M \overline{L}\}$	Sender/Message Unobservability with Receiver-Message Unlinkability
$\overline{SO}\{R M \overline{L} - P'\}$	Sender/Message Unobservability with Receiver-Message Unlinkability leaking Pseudonym
$\overline{SO}$	Sender Unobservability
$\overline{SO} - \|U\|$	Sender Unobservability leaking User Number
$\overline{SO} - H$	Sender Unobservability leaking Histogram
$\overline{SO} - P$	Sender Unobservability leaking Pseudonym
$S F \overline{L}$	Sender-Frequency Unlinkability
$S F \overline{L} - H$	Sender-Frequency Unlinkability leaking Histogram
$S F \overline{L} - P$	Sender-Frequency Unlinkability leaking Pseudonym
$S M \overline{L}$	Sender-Message Unlinkability
$S M \overline{L} - P$	Sender-Message Unlinkability leaking Pseudonym
$\overline{SO}[M\overline{O} - \|M\|]$	Sender Unobservability with Message Unobservability leaking Message Length
$(2S)\overline{L}$	Twice Sender Unlinkability
$(SM)\overline{O}$	Sender-Message Pair Unobservability
$(SM)\overline{L}$	Sender-Message Pair Unlinkability
$\overline{SO}'$	Restricted Sender Unobservability
Receiver notions	analogous
$C\overline{O}$	Communication Unobservability
$\overline{O}$	Unobservability
$(SR)\overline{O}$	Sender-Receiver Unobservability
$M\overline{O}[M\overline{L}]$	Message Unobservability with Message Unlinkability
$M\overline{O} - \|M\|$	Message Unobservability leaking Message Length
$(SR)\overline{L}$	Sender-Receiver Pair Unlinkability

Table 7. Notions

Symbol	Description
$\mathcal{A}$	Adversary
Ch	Challenger
Π	ACN protocol model
$b \in \{0,1\}$	Challenge bit
$g \in \{0,1\}$	Adversary's guess
$\underline{r}_0 = (r_{0_1}, r_{0_2}, \dots, r_{0_l})$	Batch of communications
$r_{b_i} \in \{\diamond, (u, u', m, aux)\}$	Communication
$\diamond$	Nothing is communicated
(u, u', m, aux)	m is sent from u to u' with auxiliary information aux
$(\underline{r}_{0_1}, \dots, \underline{r}_{0_k})$	(First) Scenario
$\perp$	Abort game
$\mathcal{U}$	Set of possible senders
$\mathcal{U}'$	Set of possible receivers

Table 6. Symbols used in the Game

X_1	X_2	I	attack
$\overline{SO}\{\overline{RO} - \|U'\|\}$	$(2R)\overline{L}$	$\|U'\|$	$\langle((u'_0, m_0), switchStage, (u'_0, m_0)),$ $((u'_1, m_0), switchStage, (u'_1, m_0))\rangle,$ $\langle((u'_0, m_0), switchStage, (u'_1, m_0)),$ $((u'_1, m_0), switchStage, (u'_0, m_0))\rangle$
$\overline{SO}\{\overline{RO} - P'\}$	$M\overline{O} - \|M\|$	m	$((m_2)), ((m_3))$
$\overline{SO}\{\overline{RO} - P'\}$	$(RM)\overline{O}$	$\|U'\|, m$	$\langle((u'_0, m_0), (u'_0, m_2)),$ $((u'_0, m_0), (u'_1, m_3))\rangle,$ $\langle((u'_0, m_0), (u'_0, m_3)),$ $((u'_0, m_0), (u'_1, m_2))\rangle$
$\overline{SO}\{\overline{RO} - P'\}$	$(RM)\overline{L}$	P'	$\langle((u'_0, m_2), (u'_0, m_0), (u'_1, m_1)),$ $((u'_0, m_2), (u'_1, m_1), (u'_0, m_0))\rangle,$ $\langle((u'_0, m_2), (u'_0, m_1), (u'_1, m_0)),$ $((u'_0, m_2), (u'_1, m_0), (u'_0, m_1))\rangle$

Table 8. Some counter example ideas with $X_1' = X_1$

Notion	Name	Properties
$L\overline{SO}$	Loopix's Sender Unobservavility	$E_\diamond$
$L\overline{RO}$	Loopix's Receiver Unobservability	$E_\diamond$
$\overline{SO}'$	Restricted Sender Unobservability	$\nrightarrow \wedge E_S$
$\overline{RO}'$	Restricted Receiver Unobservability	$\nrightarrow' \wedge E_R$

Table 9. Definition of the Loopix notions