Anonymity Trilemma: Strong Anonymity, Low Bandwidth Overhead, Low Latency—Choose Two

Debajyoti Das
Purdue University, USA
das48@purdue.edu

Sebastian Meiser
University College London, UK
s.meiser@ucl.ac.uk

Esfandiar Mohammadi
ETH Zurich, Switzerland
mohammadi@inf.ethz.ch

Aniket Kate
Purdue University, USA
aniket@purdue.edu

Abstract—This work investigates the fundamental constraints of anonymous communication (AC) protocols. We analyze the relationship between bandwidth overhead, latency overhead, and sender anonymity or recipient anonymity against a global passive (network-level) adversary. We confirm the trilemma that an AC protocol can only achieve two out of the following three properties: strong anonymity (i.e., anonymity up to a negligible chance), low bandwidth overhead, and low latency overhead.

We further study anonymity against a stronger global passive adversary that can additionally passively compromise some of the AC protocol nodes. For a given number of compromised nodes, we derive as a necessary constraint a relationship between bandwidth and latency overhead whose violation makes it impossible for an AC protocol to achieve strong anonymity. We analyze prominent AC protocols from the literature and depict to which extent those satisfy our necessary constraints. Our fundamental necessary constraints offer a guideline not only for improving existing AC systems but also for designing novel AC protocols with non-traditional bandwidth and latency overhead choices.

I. INTRODUCTION

Millions of users from all over the world employ anonymous communication networks, such as Tor [1], to protect their privacy over the Internet. The design choice made by the Tor network to keep the latency and bandwidth overheads small has made it highly attractive to its geographically diverse user-base. However, over the last decade, the academic literature [2]–[8] has demonstrated Tor’s vulnerability to a variety of traffic correlation attacks. In fact, Tor also has been successfully attacked in practice [9].

It is widely accepted that low-latency low-bandwidth overhead of anonymous communication (AC) protocols, such as Tor [10], can only provide a weak form of anonymity [11]. In the anonymity literature, several AC protocols were able to overcome this security barrier to provide a stronger anonymity guarantee (cryptographic indistinguishability-based anonymity [12], [13]) by either increasing the latency overhead or the bandwidth overhead. In particular, high-latency approaches (such as threshold mix networks [14]) can ensure strong anonymity by introducing significant communication delays for users messages, while high-bandwidth approaches (such as Dining Cryptographers network [15] and its extensions [16]–[18]) can provide strong anonymity by adding copious noise (or dummy) messages.

There have been a few efforts to propose hybrid approaches [19]–[24] that try to provide anonymity by simultaneously introducing latency and bandwidth overhead. However, it is not clear how to balance such system parameters to ensure strong anonymity while preserving practical performance.

In general, in the last 35 years a significant amount of research efforts have been put towards constructing novel AC protocols, deploying them, and attacking real-world AC networks. However, unlike other security fields such as cryptography, our understanding regarding the fundamental limits and requirements of AC protocols remains limited. This work takes some important steps towards answering fundamental question associated with anonymous communication. “Can we prove that strong anonymity cannot be achieved without introducing large latency or bandwidth overhead? When we wish to introduce the latency and bandwidth overheads simultaneously, do we know the overhead range values that still fall short at providing stronger anonymity?”

Our Contribution. We confirm a previously conjectured [24], [25] relationship between bandwidth overhead, latency overhead and anonymity. We find that there are fundamental bounds on sender and recipient anonymity properties [12], [13], [26], [27] of a protocol that directly depend on the introduced bandwidth and latency overheads.

This work presents a generic model of AC protocols using Petri nets [28], [29] such that different instantiations of this model will represent different AC protocols, covering most practical AC systems in the literature. We derive upper bounds on anonymity as functions of bandwidth overhead and latency overhead, against two prominent adversary classes: global passive network-level adversaries and strictly stronger adversaries that additionally (passively) compromise some protocol parties (e.g., relays in case of Tor). These bounds constitute necessary constraints for anonymity. Naturally, the constraints are valid against any stronger adversary class as well.

For both adversary classes, we analyze two different user distributions (i.e., distributions that determine at which time or rate users of the AC protocol send messages): (i) synchronized user distributions, where users globally synchronize their messages, and (ii) unsynchronized user distributions, where each user locally decides when to send his messages independent of other users.

We analyze the trade-off between latency overhead and bandwidth overhead required to achieve strong anonymity, i.e., anonymity up to a negligible (in a security parameter $\eta$) chance of failure. For any AC protocol where only a fraction
of $\beta \in [0,1]$ users send noise messages per communication round, and where messages can only remain in the network for $\ell \geq 0$ communication rounds, we find that against a global network-level adversary no protocol can achieve strong anonymity if $2\beta \ell < 1 - \frac{1}{\text{poly}(\eta)}$ even when all the protocol parties are honest. In the case where a strictly stronger adversary additionally passively compromises $c$ (out of $K$) protocol parties, we show that strong anonymity is impossible if $2(\ell - c)\beta < 1 - \frac{1}{\text{poly}(\eta)}$ (for $c < \ell$), or $2\beta \ell < 1 - \frac{1}{\text{poly}(\eta)}$ and $\ell \in \mathcal{O}(1)$ (for $c \geq \ell$).

We also assess the practical impact of our results by analyzing prominent AC protocols. Our impossibility results naturally only offer necessary constraints for anonymity, but not sufficient conditions for the AC protocol. However, these necessary constraints for sender and recipient anonymity are crucial for understanding bi-directional anonymous communication. In fact, we find that several AC protocols in the literature are asymptotically close to the suggested constraints. Moreover, designers of new AC protocols can use our necessary constraints as guidelines for avoiding bad combinations of latency and bandwidth-overhead.

**Organization.** Section II presents a detailed overview of our protocol model and our analysis. Section III formally defines the anonymity property, the game setup, and the user distributions. Section IV details our protocol model for AC protocols using timed colored Petri net, the anonymity invariant, and an ideal protocol. Section V and Section VI analyze bounds on the anonymity for the synchronized user distribution against non-compromising and partially compromised adversary respectively. Section VII and Section VIII analyze bounds on the anonymity for the unsynchronized user distribution. Section IX describes the results for recipient anonymity. Section X compares our anonymity bounds for some prominent AC protocols.

## II. Overview

### A. Formalization and Adversary Model

**AC Protocols as Petri Nets.** We define a view of AC protocols as Petri nets [28]–[30], i.e., as graphs with two types of labeled nodes: places, that store colored tokens, and transitions, that define how these tokens are sent over the graph. In our case, each colored token represents a message, places are the protocol parties that can receive, hold and send messages, and transitions describe how parties exchange and relay messages. Our model captures all AC protocols under the assumption that messages are transmitted directly, i.e., in order for Bob to receive a message from Alice, Alice has to send the message and the message (albeit relayed, delayed and graphically modified) eventually has to reach Bob. While this requirement may sound strict, as elaborated in Section IV-C, we effectively only exclude few esoteric protocols.

**User Distributions, Communication Rounds, Bandwidth Overhead, and Latency.** We consider two types of user distributions. In the first user distribution (synchronized) $N$ users send their messages in exactly $N$ rounds (see Figure 1 for notations). Per round, exactly one user sends a message. The protocol decides which users send noise messages in each round. In the second user distribution (unsynchronized) each user independently decides whether to send a message in a round using a coin flip, with a success probability $p$.

The model considers synchronous communication rounds as in [16], [17], [31], [32]. We model latency overhead $\ell$ as the number of rounds a message can be delayed by the protocol before being delivered. We formalize bandwidth overhead $\beta$ as the number of noise messages per user that the protocol can create in every round, i.e., the dummy message rate.

Our two types of user distributions cover a large array of possible scenarios. Results for our user distributions imply results for similar distributions, if a reduction proof can show that they are less favorable to the protocol.

**Adversaries.** We consider global passive non-compromising adversaries, that can observe all communication between protocol parties; and strictly stronger partially compromising (passive) adversaries, that can compromise protocol parties to learn the mapping between inputs and outputs for this party.

**Anonymity Property.** We leverage an indistinguishability-based anonymity notion for sender anonymity: the adversary has to distinguish two senders of its own choosing [12], [13].

For a security parameter $\eta$, we say that a protocol achieves strong anonymity, if the adversary’s advantage remains negligible in $\eta$.

Strong anonymity is relative to a strength $\eta$, which is bound to system parameters or analysis parameters such as the number of users or protocol parties, the latency overhead and the bandwidth overhead. These parameters typically increase as $\eta$ increases, which improves the protocol’s anonymity. Anonymity in relation to $\eta$ unifies a wide variety of possible analyses on how the anonymity bound changes with changing system parameters, and user numbers and behaviors. In particular, all our system and analysis parameters, such as the bandwidth overhead $\beta$, the latency $\ell$, or the number of compromised parties $c$ are actually functions in $\eta$. Each inequality on such parameters, e.g., $2\ell\beta < 1$, is an abbreviation for $2(\eta)\beta(\eta) < 1$ for sufficiently large $\eta$.

![Fig. 1. Notation](image-url)

1Such distributions might contain usage patterns, irregularities between users and synchronization failures that the adversary can exploit.

2In some analyses, individual parameters may reduce with increasing $\eta$, such as the bandwidth overhead per user, as the other parameters, such as the number of users, increase.
B. Brief Overview of the Proof Technique

As non-compromising adversaries are a subset of partially compromising adversaries, our proof technique for the former is a simplified case of the latter. In general, we derive our results in four main steps.

First, we define a concrete adversary $A_{paths}$, that uses a well established strategy: upon recognizing the challenge message (as soon as it reaches a receiver) $A_{paths}$ constructs the possible paths this message could have taken through the network, and tries to identify the user who has sent the message.

Second, given the concrete adversary $A_{paths}$, we identify a necessary invariant that any protocol has to fulfill in order to provide anonymity. Intuitively: both challenge users chosen by the adversary must be active (i.e., send at least one message) before the challenge message reaches the recipient, and it must be possible for these messages to meet in at least one honest party along the way. We prove that indeed this natural invariant is necessary for anonymity.

Next, we propose an ideal protocol $\Pi_{ideal}$ that is optimal in terms of satisfying the invariant: The probability that $\Pi_{ideal}$ fulfills the necessary invariant is at least as high as for any protocol within our model (limited by the same constraints for $\beta$ and $\ell$). Thus, no protocol can be better at winning against $A_{paths}$ than $\Pi_{ideal}$ is in satisfying the necessary invariant.

Finally, we calculate a bound $\delta$ on the probability of $\Pi_{ideal}$ to satisfy the necessary invariant. By calculating $\delta$, we obtain a lower bound (of $\delta$) on the adversarial advantage against all protocols within our model.\(^3\)

C. Scenarios and Lower Bounds

We devise necessary constraints for four different scenarios. Let $\Pi$ be a protocol in our model, with $N$ users, restricted by bandwidth overhead $\beta \in [0, 1]$ and latency overhead $\ell \geq 0$. For the compromising cases, the adversary can compromise $c$ out of $K$ protocol parties. We derive the following lower bounds for $\delta$-sender anonymity in the respective scenarios.

Synchronized Users, Non-compromising Adversaries:

$$\delta \geq 1 - f_{\beta}(\ell), \text{ where } f_{\beta}(x) = \min \left(1, \left(\frac{x + \beta N}{N-1}\right)\right).$$

Synchronized Users, Partially Compromising Adversaries:

$$\delta \geq \begin{cases} 1 - \left[1 - \frac{c}{K}\right] f_{\beta}(\ell) & c \geq \ell \\ 1 - \left[1 - \frac{c}{K}\right] f_{\beta}(c) - f_{\beta}(\ell - c) & c < \ell. \end{cases}$$

Unsynchronized Users, Non-compromising Adversaries:

$$\delta \geq 1 - \left[\frac{1}{2} + f_p(\ell)\right], \text{ where for } p \approx \beta \text{ we have } f_p(x) = \min(\frac{1}{2}, 1 - (1 - p)^x) \text{ for a positive integer } x.$$  

\(^3\) $A_{paths}$ is a possible adversary against all protocols within our model. If $A_{paths}$ wins whenever the invariant is not satisfied and our ideal protocol $\Pi_{ideal}$ (bounded by $\beta$ and $\ell$) is the best protocol for satisfying the invariant, then $A_{paths}$ will also have an advantage of at least $\delta$ against any protocol within our model (that is also bounded by $\beta$ and $\ell$). Thus, our bound for $\delta$ describes a lower bound on the adversarial advantage against any protocol within the model, while against particular protocols there can be other adversaries (in the same adversary class) with an even higher advantage.

Unsynchronized Users, Partially Compromising Adversaries:

$$\delta \geq \begin{cases} 1 - \left[1 - \frac{c}{K}\right] [\frac{1}{2} + f_p(\ell)] & c \geq \ell \\ \left(1 - 1 / (c^2)\right) \left[\frac{1}{2} + f_p(\ell)\right] & c < \ell. \end{cases}$$

On our project webpage [33], we visualize the lower bounds presented above using interactive 3D surface plots. In Appendix D, we present a few interesting snapshots of those lower bound plots.

We derive bounds for sender anonymity in the body of this paper. The bounds for recipient anonymity are obtained analogously and can be found in Appendix C.

D. Interpretation and Interesting Cases

Our first and third lower bounds, for respectively synchronized and unsynchronized user behaviors against in a non-compromised AC network, suggest an anonymity trilemma. Both lower bounds can be simplified under some natural constraints to the following simplified lemma:

Lemma 1 (Informal Trilemma). For security parameter $\eta$, no protocol can achieve strong anonymity if $2(1 - \epsilon) < \epsilon(\eta)$, where $\epsilon(\eta) = \frac{1}{1-\eta}$ for any positive constant $d$.

Ideal asymptotic values for latency overhead is $\ell = O(1)$ (i.e., a constant number of hop separation from the receiver), while ideal asymptotic values for bandwidth overhead is $\beta = O(1/\sqrt{N}) = O(1/poly(\eta))$ (i.e., a constant number of message per round from all $N = poly(\eta)$ users combined). It is easy to see that for this ideal overhead $\beta = O(1/poly(\eta))$, the trilemma excludes strong anonymity, while, with latency overhead $\ell = N = O(poly(\eta))$ or with bandwidth overhead $\beta = O(1)$, the trilemma does not exclude strong anonymity.

We find some interesting possible overhead constraints for strong anonymity (e.g. $\ell = O(\eta)$ and $\beta = O(1/\eta)$) demanding some compromise in both latency and bandwidth. These constraints can help understand and improve existing AC protocols as well as inform the design of future AC protocols.

For partially compromised scenarios the requirements are naturally stronger. All constraints discussed for compromised case in the following part are in addition to the requirements from the non-compromised case. While bandwidth overhead might be sufficient against non-compromising adversaries, it is not sufficient if parts of the protocol are compromised. With $\ell = \eta$ and $\frac{\ell}{\beta} = constant$ strong anonymity may be possible, whereas with $\ell = O(1)$, strong anonymity is impossible, even for $K \in poly(\eta)$ and $c = O(1)$.

In case $c < \ell$, strong anonymity guarantees may be possible only if $2(\ell - c)p > 1 - \epsilon(\eta)$, where $p = p' + \beta$ combines the genuine user messages $p'$ with their bandwidth overhead $\beta$. Our result shows a connection between the expected usage behavior $p$ and the latency $\ell$. If $p$ is not particularly large, the latency cannot be low; otherwise, the path-length cannot be sufficiently high to ensure mixing at an honest node. In other words, unless $p$ is very large (as should be the case for
and bandwidth overheads. Our necessary constraints enable protocol designers of AC protocols to avoid bad trade-offs between latency and bandwidth overhead. For a given expected user behavior and a given target attacker against which the AC protocol shall provide anonymity, our constraints clearly state which combinations of latency and bandwidth overhead to avoid.

E. Related Work

In contrast to previous work, our work provides necessary constraints for strong anonymity w.r.t. to bandwidth and latency overhead. While there is a successful line of work on provable anonymity guarantees [12], [26], [27], [34]–[37], it is incomparable since it provides lower bounds on anonymity for specific protocols only. Oya et al. [38] cast their attack in a general model and provide a sophisticated generic attacker. However, they only compute bounds w.r.t. a dummy message rate against timed pool mixes, not against other protocols and not w.r.t. latency and compromisation rate. Even more important, none of these results discuss the relationship of the lower bounds for latency and bandwidth overheads.

III. Anonymity Definition and User Distributions

This section defines the indistinguishability-based anonymity notion for which we prove our necessary constraints.

A. AnoA-Style Anonymity Definition

We define our anonymity notions with a challenge-response game following the AnoA definition [26], [27], where the challenger simulates the protocol and the adversary tries to deanonymize users. The challenger Ch(Π, α, b) allows the adversary to control user communication in the network, up to an uncertainty of one bit for challenges, and is parametric in the following parts: (i) the AC protocol Π to be analyzed, (ii) the so called anonymity function α, that describes the specific variant of anonymity such as sender anonymity, recipient anonymity and relationship anonymity, (iii) and the challenge bit b which determines the decision the challenger takes in challenge inputs from the adversary.

Given a security parameter η, we quantify the anonymity provided by the protocol Π simulated by Ch(Π, α, b) in terms of the advantage the probabilistic polynomial time (PPT) adversary A has in correctly guessing Ch’s challenge bit b. We measure this advantage in terms of indistinguishability of random variables additively, where the random variables in question represent the output of the interactions (A|Ch(Π, α, 0)) and (A|Ch(Π, α, 1)).

\textbf{Definition 1} \((\alpha, \delta)\text{-IND-ANO}\). A protocol Π is \((\alpha, \delta)\text{-IND-ANO}\) for the security parameter η, an adversary class C, an anonymity function α and a distinguishing factor \(\delta(\cdot) \geq 0\), if for all PPT machines A ∈ C,
\[
\Pr[0 = \langle A|Ch(Π, α, 0)\rangle] \leq \Pr[0 = \langle A|Ch(Π, α, 1)\rangle] + \delta(η).
\]

For an anonymity function α, we say that a protocol Π provides strong anonymity [12], [13] if it is \((α, δ)\text{-IND-ANO}\) with \(δ \leq \text{neg}(η)\) for some negligible function \text{neg}. Note that η does not measure the size of the anonymity set, but the computational limitation of the adversary.

\textbf{Sender Anonymity.} Sender anonymity characterizes the anonymity of users against a malicious server through the inability of the server (or some intermediary) to decide which of two self-chosen users have been communicating with the server. We borrow the sender anonymity \(α_{SA}\) definition from the AnoA framework [26], where \(α_{SA}\) selects one of two possible challenge users and makes sure that the users cannot be distinguished based on the chosen sender(s) or message(s).

\textbf{Definition 2} (Sender anonymity). A protocol Π provides δ-sender anonymity if it is \((α_{SA}, δ)\text{-IND-ANO}\) for \(α_{SA}\) as defined in Figure 2.

\textbf{Recipient Anonymity.} Recipient anonymity characterizes that the recipient of a communication remains anonymous, even to observers that have knowledge about the sender in question. Similar to sender anonymity, we borrow the recipient anonymity \(α_{RA}\) definition from the AnoA framework, where \(α_{RA}\) selects one of two possible recipients for a message and makes sure that the recipients cannot be distinguished based on the chosen sender(s) or message(s).

\textbf{Definition 3} (Recipient anonymity). A protocol Π provides δ-recipient anonymity if it is \((α_{RA}, δ)\text{-IND-ANO}\) for \(α_{RA}\) as defined in Figure 2.

We omit the detailed technical notation of the anonymity functions in the following sections, and write \(\Pr[0 = A \mid b = i]\) instead of \(\Pr[0 = \langle A|Ch(Π, α_{SA}, i)\rangle]\).

B. Game Setup

Let \(S\) be the set of all senders, \(R\) be the set of all recipients, and \(P\) be the set of protocol parties that participate in the execution of the protocol (like relays/mix-nodes in Tor/mix-nets, for DC-net or P2P mixing users and protocol parties are the same). We consider a system of total \(|S|\) = N senders. For sender anonymity, we need only a single element in \(R\), while for recipient anonymity we only need one in the set \(S\). For illustration, we focus on sender anonymity in the main body of the paper, discuss the adjustments for recipient anonymity in Section IX, and postpone the proof for recipient anonymity to Appendix C. We allow the adversary (for sender anonymity) to set the same entity (say \(R\)) as the recipient of all messages, and expect \(R\) to be compromised by the adversary. The adversary uses a challenge (as defined in Figure 2) of

\textsuperscript{4}AnoA also allows a multiplicative factor \(c\); we use the simplified version with \(ε = 0\), such that \(δ\) directly corresponds to the adversarial advantage.
We assume a standard (bounded) synchronous communication model as in [16], [17], [31], [32], where a protocol operates in a sequence of communication rounds. Any party can send a message directly to any other party. We consider a completely connected topology, which means any party can communicate with each other to implement broadcast or point-to-point communication links between any two parties, including users. We consider two types of adversaries: 

- **Completely Compromising Adversaries (C.C.A.):** The adversary can only observe the content of a message when it reaches its final recipient, we consider the AC protocol is non-compromised. Our strictly stronger partially compromising adversaries (which model hacking and infiltration capabilities) can additionally compromise some of the AC parties in the setup phase of the game to obtain these parties’ mapping between the input messages and output messages during the protocol’s runtime. In the case of partially compromising adversaries, we say that the AC protocol is partially compromised.

**C. User Distributions**

We consider two kinds of user distributions in our anonymity games and both of them assume an $N$ sized set $S$ of users that want to send messages. In both cases, the adversary can choose any two senders $u_0, u_1 \in S$. However, the time and method by which they actually send messages differs:

- **In the synchronized user distribution** the users globally synchronize who should send a message at which point in time. We assume that each user wants to send exactly one message. Consequently, we choose a random permutation of the set of users $S$ and the users send messages in their respective round. In every single round out of a total of $N$ rounds exactly one user sends a message. Since the users globally synchronize their sending of messages, we allow the protocol to also globally decide on the bandwidth overhead it introduces. Note that here the requirements are identical to those of the Bulk protocol in [17].

- **In the unsynchronized user distribution** each of the $N$ users wants to send messages eventually and we assume that each user locally flips a (biased) coin every round to decide whether or not to send a message. In this case we define the bandwidth overhead as an increased chance of users sending messages. Since the protocol does not globally synchronize the input messages, for noise messages also we allow the users to decide it locally and send noise messages with a certain probability.

**IV. A Protocol Model for AC Protocols**

An AC protocol allows any user in the set of users $S$ to send messages to any user in $R$, via a set of anonymizing parties $P$. We define protocols that are under observation of an eavesdropping adversary $\mathcal{A}$ that may have compromised a set of $c$ parties $P_c \subseteq P$ and that furthermore observes the communication links between any two parties, including users.

Technically, whenever a party $P_1 \in P \cup S$ sends a message to another party $P_2 \in P \cup R$, the adversary is able to observe this fact together with the current round number. However, we assume the protocol applies sufficient cryptography, s.t., the adversary cannot read the content of any message except the messages sent to the malicious recipient. Thus, the adversary can only recognize the challenge message when it reaches the recipient.

For an actual protocol, the sets $S$, $R$, and $P$ might not be mutually exclusive [15], [16], [18]. Since we have only one malicious party in $R$, and the content of a message can only be read when it reaches its final recipient, we consider $R$ to be mutually exclusive from $S \cup P$ for the purpose of simplicity.

With the above preliminaries in mind, we shall now formally define our generic AC protocol using a Petri net model.

---

5While a time-sensitive model [43] would be more accurate, e.g., for low-latency protocols like Tor [44], such a model would only strengthen the attacker. As we present necessary constraints, our results also hold for the more accurate setting.
A. Protocol Model

We model any AC protocol with K parties by a timed colored Petri net [28]–[30] \( M \), consisting of places \( S \) for the users, \( P_1, \ldots, P_K \) symbolizing the protocol parties, \( S_1 \) for randomness and \( R \) for recipients of messages, and colored tokens \( m \) symbolizing the messages (real or noise) sent by clients or protocol parties, and transitions \( T_S \) for inserting messages into the network and \( T_{P_1}, \ldots, T_{P_K} \) as functions for sending the messages from one party to another. The structure of the Petri net with its places, tokens and transitions remains the same for every AC protocol. However, the implementation of the guards within the transitions is different for different protocols; protocols can choose to which party messages are to be sent next and whether they should be delayed. Protocols in \( M \) are oblivious to the challenge message or the challenge users – and so is the implementation of the guards within the transitions. We refer to Figure 3 for a graphical depiction of Petri net model \( M \).

**Definition 4 (Colored token).** A colored token is represented by the tuple \( m = (msg, meta, t_r, ID_t, prev, next, ts) \), where,
- \( msg \) is the content of the message,
- \( meta \) is the internal protocol meta-data for this message,
- \( t_r \) is the time the message can remain in the network,
- \( ID_t \) is a new unique ID generated by each transition for each token by honest parties; dishonest parties instead keep \( ID_t \) untouched to allow the adversary to link incoming and outgoing messages,
- \( prev \) is party/user that sent the token and \( next \) is the user/protocol that receives the token.
- \( ts \) is the time remaining for the token to be eligible for a firing event (a feature of timed Petri net). Here, \( ts \) either describes when new messages are introduced into the Petri net or is set to the next round, such that messages can be processed in every round as soon as they enter the network.

Transitions in the Petri net model \( M \)

\( T_S \) on tokens \( q = (msg, \_\_\_\_, u, \_\_\_\_) \) from \( S \) and \( S_1 \) from \( S \):

\[
(P_1, meta) = f_1(q, \$); \ ID_t = \text{a fresh randomly generated ID}
\]

\( r = \text{current round} \); \( t = (msg, meta, t_r, ID_t, u, P_i, 1) \)

if \( P_i = R \) then Tokens = Tokens \( \cup \) \((msg, \_\_\_\_, ID_t, u, P_i, 1), r) \)

else Tokens = Tokens \( \cup \) \((msg, \_\_\_\_, ID_t, u, P_i, 1), r) \)

**Output:** token \( t \) at \( P_i \)

\( T_{P_i} \) on tokens \( q = (msg, t_r, ID_t, \_\_\_\_, P_i, ts) \) from \( P_i \), \( S_1 \) from \( S \):

\[
(P', meta') = f_1(q, \$); \ r = \text{current round}
\]

if \( t_r - 1 = 0 \) then \( P' = R \)

else if \( P_i \) is honest then \( ID_t' = \text{a fresh randomly generated ID} \)

else if \( P_i \) is compromised then \( ID_t' = ID_t \)

\( t = (msg, meta', t_r - 1, ID_t', P_i, P', 1) \)

if \( P_i = R \) then Tokens = Tokens \( \cup \) \((msg, \_\_\_\_, ID_t', P_i, P', 1), r) \)

else Tokens = Tokens \( \cup \) \((msg, \_\_\_\_, ID_t', P_i, P', 1), r) \)

**Output:** token \( t \) at \( P' \)

\( f_1 \): The code for this function is provided by protocol \( \Pi \). It decides to which party the message is sent next, as well as the content of the meta field in the token.

Fig. 4. Transitions in the Petri net model \( M \)

The four fields \( ID_t, prev, next, ts \) are public, and are visible to the adversary. The remaining three fields \( msg, meta \) and \( t_r \) in a token are private and can not be observed by the adversary, with the exception that \( msg \) can be observed when a message reaches its destination, i.e., is received by a recipient.\(^6\)

Formally, we introduce a set Tokens, that is initially empty and in which we collect the pair \((t, r)\), where \( t \) is a copy of a token and \( r \) the round number in which the token was observed.

**Places.** Any AC protocol with \( K \) parties \( P = \{P_1, \ldots, P_K\} \) consists of the following places:

- \( S \): A token in \( S \) denotes a user message (real or noise) which is scheduled to enter the network after \( ts \) rounds.
- \( S_1 \): This place is responsible for providing randomness. Whenever a transition picks a token from this place, the transition basically picks a random value.
- \( P_i \) with \( P_i \in P \): A token in \( P_i \) denotes a message which is currently held by the party \( P_i \in P \).
- \( R \): A token in \( R \) denotes a message which has already been delivered to a recipient.

**Transitions.** As part of the initial configuration, the challenger populates \( S \) on behalf of the protocol. All other places are initially empty. The transitions then consumes token from one place and generate tokens in other places, to modify the configuration of the Petri net. The event of consumption of a token from one place by a transition and generation of a new token represents the movement of a message from one party to another. We define the following transitions (we refer to Figure 4 for the pseudocodes of the transitions):

\(^6\)For sender anonymity we only consider one recipient and thus, for simplicity, do not need to specify the recipient in the token. For recipient anonymity, the colored token additionally has a private field for the recipient.
the set \( \Pi \) that if \( \Pi \) challenge message to a copy of the respective (new) token ID and a challenge message \( A \) adversary is executed. We consider the following game between a PPT adversary (c.f. Figure 4). However, when the field \( msg \) is the next place \( t \), the values of \( T \) are removed. If the place where \( msg \) was sent by \( u \), then additionally the field \( msg \) is removed. For each element in \( S∪\beta \). For every message that is sent from one party in \( S \) and a latency overhead \( t \) and a latency overhead \( T \). In either case, the transition also adds an element \( (msg,\beta,\ldots,u,\ldots,ts) \) from \( P \) and a token from \( S \) to write \( t = (msg,\beta,\ldots,u,P,ts = 1) \) to \( P \); if the values of \( i \) and meta are decided by the AC protocol. \( T \) should use the transition \( T \) with the protocol specific implementation of \( f \).

**Run protocol \( \Pi \) on \( r = (u^*,R^*,m^*) \) and user distribution \( U \):**

\( I_U \leftarrow \Pi_{\text{wrapper}}(r,U) \),

where \( I_U \) is a set and each element in \( I_U \) is a tuple \( (u,R,m,ts) \). Run \( \Pi_{\text{core}}(I_U,U,\beta) \).

**\( \Pi_{\text{wrapper}} \) on \( r = (u^*,R^*,m^*) \) and user distribution \( U \):**

- generate a set \( I_U \) following \( U \),
- where \( I_U \) is a set and each element in \( I_U \) is a tuple \( (u,R,m,ts) \). \( e = (u^*,R^*,m^*,ts^*) \) \( \leftarrow I_U \) such that \( u^* = u^* \land R^* = R^* \).
- if \( e \) exists then
  \( I_U \leftarrow \{(u^*,R^*,m^*,ts^*)\} \cup I_U \setminus \{e\} \)
- end if
- for each element \( e = (u,R,m,ts) \) in \( I_U \) do
  - Add a token \( t = (msg,\beta,\ldots,u,\ldots,ts) \) in the place \( S \).
- end for

**Output:** \( I_U \)

Add tokens in the place \( S \) within the limit of \( \beta \) noise messages per user per round.

Run the petri net with the protocol specific implementation of \( f \).

**Fig. 5. Description of protocol \( \Pi \)**

**B. Game Setting**

We use the sender anonymity notion from the AnoA framework [27]. To increase readability, we summarize the AnoA definition and explicitly write down how the sender anonymity notion \( \alpha_{SA} \) works and how the wrapper around the protocol is executed. We consider the following game between a PPT adversary \( A \) and an honest challenger \( Ch(\Pi,\alpha_{SA},b) \):

- \( A \) compromises up to \( c \) parties from \( P \).
- \( A \) chooses two distinct users \( u_0 \) and \( u_1 \). \( A \) sends a challenge message \((\text{Challenge},u_0,u_1,R,R,m^*)\) for those chosen users.
- \( Ch \) then runs protocol \( \Pi \) on \((u_0,R,m^*)\). \( \Pi \) is executed in two parts, \( \Pi_{\text{wrapper}} \) and \( \Pi_{\text{core}} \), as described below. (We refer to Figure 5 for the pseudocode of \( \Pi \)).
- First, \( \Pi_{\text{wrapper}} \) generates tokens following the user distribution and embeds the challenge message \((u_0,R,m^*)\) in the tokens. \( \Pi_{\text{wrapper}} \) adds all the tokens to the place \( S \). \( \Pi_{\text{wrapper}} \) does not pass any information about the challenge user or the challenge message to \( \Pi_{\text{core}} \).
- We also allow \( \Pi_{\text{core}} \) to add noise tokens to \( S \), limited by the protocol’s restrictions on bandwidth overhead. After that, \( \Pi_{\text{core}} \) runs the petri net with protocol specific implementation of the transitions.
- For each element in \( T \), \( A \) can see the round number as well as the public parts of the token (\( ID_{\text{t}}, \text{prev}, \text{next}, ts \)), but the private parts (\( msg, \text{meta}, ts \)) are not communicated to the adversary (c.f. Figure 4). However, when the field \( \text{next} \) is the recipient, the \( msg \) field is not obfuscated.
- The goal of the adversary is to deanonymize the sender of the challenge message, i.e., to learn whether the challenge message was sent by \( u_0 \) or by \( u_1 \). The interaction between \( Ch \) and \( A \) ends as soon as \( A \) makes a guess.

**Validity of the Protocol Model.** Protocols in the above protocol model behave as expected (more details in Lemma 2 in Appendix A). We show in Lemma 2 that the protocols indeed have a bandwidth overhead of \( \beta \) and a latency overhead of \( \ell \). For every message that is sent from one party in \( S \cup P \) to another party in \( P \cup R \), the adversary learns the time, the sender, and the receiver. When a message leaves the network, the attacker learns whether it was the target (i.e., the challenge)

**Message.** The attacker also learns the mapping between the input and output messages of compromised parties. For recipient anonymity, the attacker instead learns whether a message is the target (i.e., challenge) message after it leaves the sender.

**C. Expressing Protocols**

Our protocol model \( M \) allows the expression of any AC protocol with very few, esoteric exceptions.

Mix networks can be naturally embedded into our model, in particular any stop-and-go mix [46] that uses discrete distribution and even AC protocols with specialized path selection algorithms [47], [48]. For the sake of our necessary constraints, low-latency protocols (with time-bounded channels) that are not round-based (e.g., Tor [44]) can be expressed in a round-based variant, since it only strengthens the protocols anonymity properties. This section illustrates embedding techniques into our model for some other kinds of protocols, but a much larger variety of protocols can be expressed in our model.

**Users as protocol parties.** In peer-to-peer protocols like dining cryptographers networks (DC net) [16], [18], users act as a type of relays. Also, any noise sent by users counts into the bandwidth overhead of the protocol (we will see in Claim 2 that noise sent by nodes that are not users can be treated differently). Whenever a user wants to send a message it should use the transition \( T_S \), but when it acts as a relay it should use the transition \( T_P \). For interested readers, we show in Appendix A how to model a specific DC net type protocol using our Petri net model.

**Splitting and Recombining Messages.** We model protocols that split and later re-combine messages by declaring one of
the parts as the main message and the other parts as noise, which may count into the bandwidth overhead. This declaration is mainly required for the analysis, i.e., for evaluating the success of the adversary and for quantifying the amount of noise messages introduced by the protocol. We do not restrict the strategy by which the protocol decides which message is “the main share” (i.e., the message that is sent on) and which is “an additional share” (i.e., a fresh noise message). A more complex scenario involves threshold schemes in which a smaller number of shares suffices for reconstructing the message and in which some shares are dropped randomly. In such cases we consider the protocol to decide beforehand which of the constructed shares will be dropped later and to declare one of the remaining shares the “main share”.

Broadcasting Messages. If the protocol chooses to copy or broadcast messages to several receivers, we consider the copy sent to the challenge receiver to be the main message and copies sent to other receivers to be noise (which, if the copies are created by nodes that are not users, will not count into the bandwidth overhead).7

Private Information Retrieval. In schemes based on private information retrieval we require that the receiver retrieves the information sufficiently fast (within the latency limit). Otherwise, our method is similar to the broadcasting of messages: the receiver of interest will retrieve the main message, whereas other receivers will retrieve copies that are modeled as noise.

Excluded Protocols. For this work we exclude protocols that cannot guarantee the delivery of a message within the given latency bound (except if this occurs with a negligible probability). Moreover, we cannot easily express the exploitation of side channels to transfer information, e.g., sending information about one message in the meta-data of another message, or sending bits of information by not sending a message.

D. Construction of a Concrete Adversary

Given two challenge users $u_0$ and $u_1$ and the set of observed tokens $(t, r) \in \text{Tokens}$, where $t$ is the token and $r$ the round in which the token was observed, an adversary can construct the sets $S_j$ (for $j \in \{0, 1\}$). Assume the challenge message arrives at the receiver $R$ in a round $r$. We construct possible paths of varying length $k$, s.t., each element $p \in S_j$ represents a possible path of the challenge message starting from $u_j$ ($j \in \{0, 1\}$) and the challenge message then arrives at $R$ in round $r_k = r$. With challenge bit $b$, $S_b$ cannot be empty, as the actual path taken by the challenge message to reach $R$ has to be one element in $S_b$.

7We note that in some cases, where users act as nodes and broadcast messages to other users, our quantification of the bandwidth overhead might be a bit harsh. If the group of users to which the broadcast will be sent is known in advance (i.e., if messages are broadcast to all users or to pre-existing groups of users), we can allow the protocol to use a single receiver for these messages instead.

$$S_j = \{ p = (t_1, \text{prev}, \ldots, t_k, \text{prev}, t_k, \text{next}) :$$

$$(t_1, r_1), \ldots, (t_k, r_k) \in \text{Tokens} \text{ s.t.}$$

$t_1, \text{prev} = u_j \land t_k, \text{next} = R$

$\land t_k, \text{msg} = \text{Challenge} \land k \leq \ell$

$\land \forall i \in \{1, \ldots, k-1\} (t_i, \text{next} = t_{i+1}, \text{prev} \land r_{i+1} = r_i + 1)$

$\land (\exists t_{i+1} : (t_{i+1}, r_{i+1}) \in \text{Tokens} \land t_{i+1}, \text{prev} = t_i, \text{next} \land t_{i+1}, \text{ID}_l = t_i, \text{ID}_l \Rightarrow t_{i+1} = t_i + 1) \}$

Definition 5 (Adversary $A_{\text{paths}}$). Given a set of users $S$, a set of protocol parties $P$ of size $K$, and a number of possibly compromised nodes $c$, the adversary $A_{\text{paths}}$ proceeds as follows: 1) $A_{\text{paths}}$ selects and compromises $c$ different parties from $P$ uniformly at random. 2) $A_{\text{paths}}$ chooses two challenge users $u_0, u_1 \in S$ uniformly at random. 3) $A_{\text{paths}}$ makes observations and, based upon those, constructs the sets $S_0$ and $S_1$. For any $i \in \{0, 1\}$, if $S_i = \emptyset$, then $A_{\text{paths}}$ returns $1 - i$. Otherwise, it returns $0$ or $1$ uniformly at random.

$A_{\text{paths}}$ thus checks whether both challenge users could have sent the challenge message. We explicitly ignore differences in probabilities of the challenge users having sent the challenge message, as those probabilities can be protocol specific. Naturally, when $c = 0$, $A_{\text{paths}}$ represents a non-compromising adversary; but when $c \neq 0$, $A_{\text{paths}}$ is partially compromising.

E. Protocol Invariants

We now investigate the robustness of protocols against our adversary. We define an invariant that, if not satisfied, allows $A_{\text{paths}}$ to win against any protocol. Moreover, we present a protocol that maximizes the probability of fulfilling the invariant.

 Necessary invariant for protocol anonymity. It is necessary that at least both challenge users send messages in one of the $\ell$ rounds before the challenge message reaches the recipient, as otherwise there is no way both of them could have sent the challenge message. Moreover, on the path of the actual challenge message, there needs to be at least one honest ( uncompromised) party, as otherwise the adversary can track the challenge message from the sender to the recipient ( $S_b$ will have exactly one element and $S_{1-b}$ will be empty). Those two conditions together form our necessary protocol invariant.

Invariant 1. Let $u_0$ and $u_1$ be the challenge users; let $b$ be the challenge bit; and let $t_0$ be the time when $u_0$ sends the challenge message. Assume that the challenge message reaches the recipient at $r$. Assume furthermore that $u_{1-b}$ sends her messages (including noise messages) at $V = \{t_1, t_2, t_3, \ldots, t_k\}$. Now, let $T = \{ t : t \in V \land (r - \ell) \leq t < r \}$. Then,

(i) the set $T$ is not empty, and

(ii) the challenge message passes through at least one honest node at some time $t'$ such that, $t' \in \{ \min(T), \ldots, r - 1\}$.

Claim 1 (Invariant 1 is necessary for anonymity). Let $\Pi$ be any protocol $\in M$ with latency overhead $\ell$ and bandwidth overhead $\beta$. Let $u_0, u_1, b$ and $T$ be defined as in Invariant 1. If Invariant 1 is not satisfied by $\Pi$, then our adversary $A_{\text{paths}}$ as in Definition 5 wins.
We next claim that it suffices to consider noise messages sent by users that also remain within the system for at most $\ell$ rounds, i.e., noise messages that follow the same rules as real messages. Note that we consider every new message originating from any user’s client as a fresh noise message.

**Claim 2** (Internal noise does not influence Invariant 1). Any message not originating from an end user $u \in S$ does not influence the probability for Invariant 1 being true. Moreover, noise messages do not contribute to the probability for Invariant 1 being true after they stayed in the network for $\ell$ rounds.

**Proof.** Let $u_0, u_1$ be the challenge users and let $b$ be the challenge bit and let $r$ be the round in which the challenge message is delivered to the recipient. We discuss both parts of the invariant separately:

(i) The set $T$ is not empty. Since by definition, $T$ is the set of messages sent by $u_{1-b}$, messages originating in any party not in $S$ do not influence $T$. Moreover, any message sent by $u_{1-b}$ in a round previous to $r-\ell$ does not influence $T$ either. Thus, noise messages staying in the protocol for more than $\ell$ rounds, does not improve the probability of $T$ being not empty.

(ii) The challenge message passes through at least one honest node at some time $t'$ such that, $t' \in \{\min(T), \ldots, r-1\}$. Obviously this second part of the invariant does not depend on any noise message. \hfill $\square$

Consequently, noise introduced by $u$ in $P$ but not in $S$ do not modify the probability to fulfill Invariant 1. We henceforth consider noise messages as a protocol input.

**F. Ideal Protocol**

We construct a protocol $\Pi_{ideal}$ that maximizes the probability of fulfilling Invariant 1. Claim 1 shows that for any protocol in our model $A_{\text{paths}}$ wins whenever Invariant 1 does not hold. Thus, an upper bound on the probability that $\Pi_{ideal}$ satisfies Invariant 1 yields an upper bound for all these protocols.

Given the set of all protocol parties $P = \{P_0, \ldots, P_{K-1}\}$ of size $K$, the strategy of $\Pi_{ideal}$ is as follows: in a round $r$, $\Pi_{ideal}$ delivers all messages scheduled for delivery to a recipient. All other messages (including the messages that enter $\Pi_{ideal}$ in round $r$) are sent to the protocol party $P_i$ with $i = r \mod K$. For every message that enters the protocol, $\Pi_{ideal}$ queries an oracle $O$ for the number of rounds the message should remain in the protocol. We define the following events:

- $u$.sent$(x, y)$: user $u$ has sent at least one message within rounds from $x$ to $y$. For a single round we use $u$.sent$(x)$.
- $\text{Cmpr}(x)$: $A_{\text{paths}}$ has compromised the next $x$ consecutive parties on the path.
- $\neg H$: NOT of event $H$.

Given a message sent at $t_0$ by sender $x$, and delivered to the recipient at $(t_0 + t)$, we define $P_t$ for sender $v \in S \setminus \{x\}$:

$$P_t = \sum_{j=t_0}^{t_0 + t} \Pr[v \text{.sent}(j) \land \neg v \text{.sent}(j + 1, t_0)] \times \Pr[\neg \text{Cmpr}(t)]$$

$$+ \sum_{j=t_0+1}^{t} \Pr[v \text{.sent}(j) \land \neg v \text{.sent}(r - \ell, j - 1)]$$

$$\times \Pr[\neg \text{Cmpr}(r-j)]$$

When $v = u_{1-b}$, and the message is the challenge message, $P_t$ is the probability of fulfilling Invariant 1, for the strategy above. For each message, oracle $O$ chooses an optimal $t$ that maximizes the expectation of $P_t$ over all users. Due to the over-approximation with this (not realizable) oracle, the resulting protocol is optimal w.r.t. Invariant 1 (refer to Claim 3).

**Claim 3** (Ideal protocol is ideal for the invariant). Against the given adversary $A_{\text{paths}}$, $\Pi_{ideal}$ satisfies Invariant 1 with probability at least as high as any other protocol in $M$.

**Proof.** We want to prove our claim by contradiction. Suppose, $\Pi_{ideal}$ is not the best protocol. That means, there exists a protocol $\Pi_{new}$, which satisfies Invariant 1 with a higher probability than $\Pi_{ideal}$, against the adversary $A_{\text{paths}}$.

Now we construct a new protocol $\Pi_{hybrid}$, which exactly follows the strategy of $\Pi_{ideal}$ with one exception: for a given message $\Pi_{hybrid}$ selects the time delay $t$ same as $\Pi_{new}$, instead of querying it from oracle $O$. Suppose, the challenge message is delivered to the recipient at round $r$. Given the set $\{\min(T), \ldots, r-1\}$, the ideal strategy for ensuring that at least one honest party is on the path of the challenge message is to ensure that as many distinct parties as possible are on this path. Also, given the time delay $t$, the value of $\min(T)$ is independent of the protocol, since protocols in $M$ are oblivious to the challenge users and the challenge message. Hence, $\Pi_{hybrid}$ has a probability of satisfying Invariant 1 at least as high as $\Pi_{new}$.

Now, if we compare $\Pi_{hybrid}$ and $\Pi_{ideal}$: they follow the same strategy. But $\Pi_{ideal}$ picks the time delay $t$ for any
message from oracle $O$ such that $t$ is optimal. The time delay $t$ can be picked for each message independent of the time delays of other messages. Hence, the value of $t$ received from oracle $O$ for the challenge message is optimal. Hence, $\Pi_{\text{ideal}}$ satisfies Invariant 1 with probability at least as high as $\Pi_{\text{hybrid}}$. Thus, $\Pi_{\text{new}}$ does not satisfy Invariant 1 with a higher probability than $\Pi_{\text{ideal}}$.\hfill \square

Note that $\Pr\left[ i = A_{\text{paths}} \mid b = i \right] \geq \frac{1}{2}$ is always true, since our adversary always guesses unless it is sure to win. So, if exactly one of $S_2$ and $S_1$ is non-empty, $A_{\text{paths}}$ certainly wins, otherwise it wins with probability $\frac{1}{2}$. The above fact, along with Claim 1 and Claim 3, helps us conclude that the best chance for any protocol against $A_{\text{paths}}$ is bounded by the probability of $\Pi_{\text{ideal}}$ satisfying Invariant 1.

V. SYNCHRONIZED USERS WITH NON-COMPROMISING ADVERSARIES

Our first scenario is a protocol-friendly user distribution $U_B$, where inputs from all users are globally synchronized: over the course of $N$ rounds, exactly one user per round sends a message, following a random permutation that assigns one round to each user. Analogously, the protocol globally instructs the users to send up to $\beta$ messages per user per round, or $B = \beta N$ noise messages per round in total.

In real life, the user distribution is independent of the protocol. However, to make the user distribution protocol-friendly in our modeling we consider a globally controlled user distribution. For this scenario, we consider non-compromising passive adversaries that can observe all network traffic.

A. LOWER BOUND ON ADVERSARIAL ADVANTAGE

**Theorem 1.** For user distribution $U_B$, no protocol $\Pi \in M$ can provide $\delta$-sender anonymity, for any $\delta < 1 - f_\beta(\ell)$, where $f_\beta(x) = \min(1, ((x + \beta N)N)/(N - 1))$.

**Proof.** By Claim 3, we know that $\Pi_{\text{ideal}}$ is an optimal protocol for satisfying Invariant 1 and by Claim 1 we know that satisfying Invariant 1 is necessary for anonymity, as otherwise our adversary $A_{\text{paths}}$ can win against the protocol. Thus, the probability that $\Pi_{\text{ideal}}$ satisfies Invariant 1 directly provides a lower bound of the adversary’s advantage against any protocol.

Let, $u_0$ and $u_1$ be the users chosen by the adversary and let $b$ be the challenge bit. Let $t_0$ be the round in which $u_b$ sends the challenge message and let $r$ be the round in which the challenge message reaches the recipient.

Recall that Invariant 1 is necessary for the protocol to provide anonymity: $u_{1-b}$ sends her messages (can be a noise message) at $V = \{t_1, t_2, t_3, \ldots, t_k\}$, then $T = \{t : t \in V \land (r - \ell) \leq t < r\}$. Since we are considering a non-compromising adversary, $\Pr[\text{Invariant 1 is true}] = \Pr[T \text{ is not empty}]$. With the above in mind, let us define the following events:

$H_1$: In $\ell$ rounds $u_{1-b}$ sends at least one noise message.

$H_2$: $u_{1-b}$ sends his own message within the chosen $\ell$ rounds.

$H_3$: there is at least one message from $u_{1-b}$ within the chosen $\ell$ rounds $\iff T$ is not empty $\iff$ Invariant 1 is true.

Consider any slice of $\ell$ rounds around the challenge message, there are exactly $(\ell - 1)$ user messages other than the challenge message. Hence, any slice of $\ell$ rounds yields the same probability of containing a user message from $u_{1-b}$, except when $r < \ell$ OR $r > N$ where the probability is smaller. Thus, no matter what value of $\ell$ is returned by $O$, $\Pr[H_2] \leq \frac{\ell - 1}{N - 1}$.

Given any values $\ell, \beta \geq 0$, $A_{\text{paths}}$ has the least chance of winning, if for a given interval of $\ell$ rounds, $\beta N$ unique users are picked to send the noise messages in such a way that they are not scheduled to send their own messages in that interval. Since, $\Pi_{\text{ideal}}$ needs only one message from $u_{1-b}$ in the interval of $\ell$ rounds for Invariant 1 to hold, it tries to maximize the number of users that send messages in that interval. Hence, $\Pr[H_1] \leq \frac{\beta N}{\ell - 1}$. Therefore,

$$\Pr[\neg H_3] = \Pr[\neg H_1, \neg H_2] \geq \max(0, (N - \ell - \beta N)/(N - 1)),$$

$$\Pr[H_3] = 1 - \Pr[\neg H_3] \leq \min(1, ((\ell + \beta N)/N - 1)).$$

Thus, we can bound the probability for the adversary as $\Pr[0 = A_{\text{paths}} | b = 1] = \Pr[1 = A_{\text{paths}} | b = 0] \leq \frac{1}{2} \Pr[H_3]$; and $\Pr[0 = A_{\text{paths}} | b = 0] \geq 1 - \frac{1}{2} \Pr[H_3]$. And therefore, since $\delta \geq \Pr[0 = A_{\text{paths}} | b = 0] - \Pr[0 = A_{\text{paths}} | b = 1]$, $\delta \geq 1 - \Pr[H_3] \geq 1 - f_\beta(\ell)$.

**B. IMPUNITY FOR STRONG ANONYMITY**

We now investigate under which constraints for $\ell$ and $\beta$ Theorem 1 rules out strong anonymity.

**Theorem 2.** For user distribution $U_B$ with $\ell < N$ and $\beta N \geq 1$, no protocol $\Pi \in M$ can achieve strong anonymity if $2\ell \beta < 1 - \epsilon(\eta)$, where $\epsilon(\eta) = \frac{1}{n^3}$ for a positive constant $d$.

**Proof.** For strong anonymity, we require: $\delta(\eta) = \neg(\eta)$, and we know that for $\Pi_{\text{ideal}}$ we have: $\delta(\eta) \geq 1 - f_\beta(\ell) = \left(\frac{N - \ell - \beta N}{N - 1}\right) \geq \left(\frac{N - \ell - \beta N}{N}\right) \geq 1 - \frac{\ell}{N} - \beta \ell$.

We assume for contradiction that there is a protocol limited by $\ell$ and $\beta$ such that $2\ell \beta < 1 - \epsilon(\eta)$ that still achieves strong anonymity. Since $\delta(\eta) = \neg(\eta)$, we know that $\epsilon(\eta) > \delta(\eta)$.

$$\epsilon(\eta) > \delta(\eta) \implies \epsilon(\eta) > 1 - \frac{\ell}{N} - \beta \ell$$

$$\implies \epsilon(\eta) > 1 - \frac{\ell}{N} - \frac{1}{2} (1 - \epsilon(\eta))$$

$$\implies 2\ell > N (1 - \epsilon(\eta)) \frac{N \delta \geq 1}{\beta \beta} \implies 2\ell > 1 - \epsilon(\eta)$$

The above contradicts the assumption that $2\ell \beta < 1 - \epsilon(\eta)$.

Note: In case $\beta N < 1$, no noise messages are allowed per round (i.e., $\beta = 0$) and thus $\delta(\eta) \geq 1 - \ell/N$, which is not negligible unless $\ell = N$, since $N = \text{poly}(\eta)$.

Note that this is a necessary constraint for anonymity, but not a sufficient condition. There can exist $\ell$ and $\beta$ such that $2\ell \beta > 1 - \neg(\eta)$, but $\Pi_{\text{ideal}}$ can not achieve strong anonymity, and hence no protocol can achieve strong anonymity. We have discussed few such examples later in the following part of this section.

**Interesting Cases.** For illustration, we now discuss a few examples for different values of $\ell$, $\beta$, and $N$. 

10
1) If $\ell = N$, we can have $\delta = 0$ even for $\beta = 0$. Anonymity can be achieved trivially by accumulating all messages from all $N$ users and delivering them together at round $(N + 1)$. In this case $2(\beta) = 0 < 1 - \epsilon(\eta)$, but also $\beta N = 0 < 1$.

2) $\beta = \frac{1}{N}, \ell = \eta$: We have $\delta \geq \frac{N - \eta - 1}{N} = \frac{1}{2} - \frac{\eta}{N}$.

In this case, strong anonymity is possible if $\frac{1}{2} - \frac{\eta}{N} < 1 - \epsilon(\eta)$. Even though $2\beta = 1 > 1 - \epsilon(\eta)$, anonymity depends on the relation between $\eta$ and $N$.

3) $\beta = \frac{1}{2}, \ell = \eta$: Here we have, $\delta \geq \frac{N - \eta - 1}{N} = \frac{1}{2} - \frac{\eta}{N}$.

4) $\beta = \frac{1}{2}, \ell = 1$: We have $\delta \geq \frac{N - 1}{N^2} \approx \frac{1}{2}$. In a best scenario, only half of the users send messages in $\ell$ rounds. Therefore, protocols cannot achieve strong anonymity here even though $2\beta > 1 - \epsilon(\eta)$.

5) $\beta = \frac{1}{N}, \ell = 3$: For $\eta > 3$ and $N > 4$, which is a very natural assumption, we have $2(\beta) = \frac{3}{N} < 1 - \epsilon(\eta)$. Then, $\delta \geq \frac{N - 3 - \frac{3}{N}}{N} > \eta(\eta)$. Then, $\delta \geq \frac{N - 3 - \frac{3}{N}}{N} = 1 - \frac{3}{N}$. Here, $\delta$ cannot be $neg(\eta)$. If we consider our $\Pi_{ideal}$, in $\ell$ rounds it receives only $(\frac{3}{N} + 3)$ messages (noise + user messages). So a maximum of $(\frac{3}{N} + 2)$ users can send messages other than the challenge user, and there is a high probability that $u_{1-b}$ has not sent a message. Hence $\Pi_{ideal}$ cannot achieve strong anonymity, and analogously no other protocol in $M$ can achieve that.

VI. SYNCHRONIZED USERS WITH PARTIALLY COMPROMISING ADVERSARIES

We now extend our analysis of the previous section by having compromised protocol parties. Given the set of protocol parties $P$, now our adversary $A_{paths}$ can compromise a set of $c$ parties $P_c \subset P$. If $A_{paths}$ can compromise all the parties in $P$, anonymity is broken trivially - that’s why we do not analyze that case separately. Recall from Section IV-D that $A_{paths}$ picks the $c$ parties from $P$ uniformly at random. We consider the same user distribution $U_B$ as in Section V.

A. Lower Bound on Adversarial Advantage

In our protocol $\Pi_{ideal}$ the oracle $O$ decides on the time $t$ to deliver each message, which is within $[1, \ell]$, s.t. $t$ maximizes the probability that Invariant I is true. Similar to Section V, we now calculate a bound on the probability that $\Pi_{ideal}$ satisfies Invariant I.

**Theorem 3.** For user distribution $U_B$, no protocol $\Pi \in M$ can provide $\delta$-sender anonymity, for any

$$\delta \leq \begin{cases} 1 - \frac{(\ell)}{K} f_{\beta}(\ell) & c \geq \ell \\ 1 - \frac{1}{K} f_{\beta}(c) - f_{\beta}(\ell - c) & c < \ell \\ \end{cases}$$

where $f_{\beta}(x) = \min(1, ((x + \beta N x)/(N - 1)))$.

**Proof.** Let $u_0, u_1$ be the challenge users and let $b$ be the challenge bit. Moreover, let $t_0$ be the time the challenge message is sent by $u_0$ and let $r = t_0 + t$ be the time it is received by the recipient, where $t$ is the delivery time decided by the oracle $O$.

We distinguish two cases, depending on $\ell$ and $c$:

1) Case $c \geq \ell$. We know, $\ell \geq t$ holds by definition. The invariant is true only if $u_{1-b}$ sends at least one message in one of the rounds between $(r - \ell)$ and $(r - 1)$. Additionally, if $u_{1-b}$ sends at least one message in $(r - \ell,..., t_0)$, the invariant holds only if there is at least one non-compromised party on the path between $t_0$ and $(r - 1)$. Whereas, if $u_{1-b}$ does not send any message in $(r - \ell,..., t_0)$, and the first message from $u_{1-b}$ in the interval $(t_0 + 1, r - 1)$ arrives at $t_1$, the invariant holds only if there is at least one non-compromised party on the path between $t_1$ and $(r - 1)$.

Note that $K > c \geq \ell$. Also recall from Section IV that $A_{paths}$ picks the $c$ parties uniformly at random from $K$ parties. Let $Pr[\text{Invariant I is true}]$

$$\leq \sum_{j=t_0}^{t_0} Pr[\text{u_{1-b}.send}(j) \land \neg \text{u_{1-b}.send}(j + 1, t_0)] \times Pr[\neg \text{Cmp}(\ell)]$$

$$+ \sum_{j=t_0+1}^{r} Pr[\text{u_{1-b}.send}(j) \land \neg \text{u_{1-b}.send}(r - \ell, j - 1)] \times Pr[\neg \text{Cmp}(r - j)]$$

$$\leq Pr[\neg \text{Cmp}(\ell)] \times Pr[\text{u_{1-b}.send}(r - \ell, r - 1)]$$

$$\leq (1 - \frac{(\ell)}{K}) \times \min(1, ((\ell + \beta N \ell)/(N - 1)))$$
By Claim 1 the adversary wins whenever Invariant 1 is not true. Hence, we know that the probability that the adversary guesses incorrectly is bounded by:
\[
\Pr \left[ \text{Invariant 1 is true} \right] \leq \frac{1}{2} \Pr \left[ 0 = A_{\text{paths}}[b = 1] = 1 \right] + \min(1, (\frac{e + \beta N}{N - 1})).
\]
Thus, \(\delta \geq 1 - \left( \frac{1}{\beta} \right) \left( \frac{\gamma}{\ell} \right) \) and \(c \geq 1 - \left( \frac{1}{\beta} \right) \left( \frac{\gamma}{\ell} \right) \).  

### 2) Case \(c \leq \ell\):

The probability that all parties on the mutual path of the challenge message and a message from the alternative sender \(u_{1-2}\) are compromised now mainly depends on the arrival time of the messages from \(u_{1-2}\). We find two sub-cases depending on the oracle’s choice for \(t\).

#### 2a) Case \(c \leq \ell\):

\[
\Pr[\text{Invariant 1 is true}] \leq \Pr[\text{time message from } u_{1-2} \text{ in } [(r - \ell), (r - c)] + \Pr[\text{time message from } u_{1-2} \text{ in } [(r - c), r]]
\]

\[
\times \Pr[\text{time message from } u_{1-2} \text{ in } [(r - c), r]] + \Pr[\text{time message from } u_{1-2} \text{ in } [(r - c), r]]
\]

\[
\times \Pr[\text{time message from } u_{1-2} \text{ in } [(r - c), r]] + \Pr[\text{time message from } u_{1-2} \text{ in } [(r - c), r]]
\]

\[
\times \Pr[\text{time message from } u_{1-2} \text{ in } [(r - c), r]] + \Pr[\text{time message from } u_{1-2} \text{ in } [(r - c), r]]
\]

\[
\times \Pr[\text{time message from } u_{1-2} \text{ in } [(r - c), r]] + \Pr[\text{time message from } u_{1-2} \text{ in } [(r - c), r]]
\]

#### 2b) Case \(t < c\):

The above expression for the probability of the adversary winning follows analogously.

### B. Impossibility for Strong Anonymity

#### Theorem 4.

For user distribution \(U_B\) with \(K \in \text{poly}(\eta)\), \(K > c \geq 1\), \(N < \eta\) and \(\beta N \geq 1\), no protocol \(\Pi \in M\) can achieve strong anonymity if \(2\ell \beta < 1 - \epsilon(\eta)\) or \(\ell \in \Omega(1)\), where \(\epsilon(\eta) = 1/\eta^4\) for a positive constant \(\epsilon\).

**Proof.** When \(c \geq \ell\): \(\delta \geq 1 - \left( \frac{1}{\beta} \right) \left( \frac{\gamma}{\ell} \right) f_{\beta}(\ell)\).  

For \(\delta\) to become \(\text{neg}(\eta)\), we need both \(1 - \left( \frac{1}{\beta} \right) \left( \frac{\gamma}{\ell} \right) f_{\beta}(\ell)\) and \(f_{\beta}(\ell)\) to become overwhelming. From Theorem 2 and Theorem 1, we know that \(2\ell \beta > 1 - \text{neg}(\eta)\) is a necessary condition for \(f_{\beta}(\ell)\) to become overwhelmingly. Now, we are left with the factor \(1 - \left( \frac{1}{\beta} \right) \left( \frac{\gamma}{\ell} \right) f_{\beta}(\ell)\). This can become overwhelming iff \(\left( \frac{\gamma}{\ell} \right) f_{\beta}(\ell)\) becomes negligible. We know that \(K > c \geq \ell\) and \(K \in \text{poly}(\eta)\). Hence, for some constant \(x\),

\[
\frac{c - \ell}{K - \ell} > \frac{1}{\eta^x} \iff \left( \frac{c - \ell}{K - \ell} \right)^x > \left( \frac{1}{\eta^x} \right)^x
\]

\[
\implies \left( \frac{c - \ell}{K - \ell} \right)^x > \frac{1}{\eta^x}
\]

For any \(\ell \in \Omega(1)\), \((1/\eta^x)^x\) is non-negligible.

To achieve strong anonymity against \(A_{\text{paths}}\), we need \(\ell \in \Omega(1)\), additional to the constraint of \(2\beta \ell > 1 - \text{neg}(\eta)\). We now focus on the constraint \(\ell \in \Omega(1)\) and refer to Section V-B for a comprehensive case study on the other constraint.

#### Interesting Cases.

Now we are going to discuss a few interesting cases for different values of \(\ell < c\), and \(K\).

1) \(\ell = c = \gamma = \text{constant}\):

In this case we have, \(\left( \frac{\gamma}{\ell} \right) = \frac{\gamma}{c} = c/\ell\). Hence, \(\left( \frac{\gamma}{\ell} \right)\) becomes negligible and strong anonymity is possible. Even though \(c\) has a high value, because of the high value of \(\ell\), there is a significant possibility that the challenge message will meet a message from \(u_{1-2}\) at some honest node, given a high value of \(\beta\) such that \(2\ell \beta > 1 - \text{neg}(\eta)\).

2) \(\ell = c = K = 1\):

Now we have, \(\left( \frac{\gamma}{\ell} \right) = \frac{\gamma}{c} = c/\ell\). Hence, \(\left( \frac{\gamma}{\ell} \right)\) is non-negligible, and hence \(\left( \frac{\gamma}{\ell} \right)\) is also non-negligible. Even though \(c\) has a small value, \(\ell\) is also small. Hence, it is unlikely that the challenge message will mix with a message from \(u_{1-2}\) at some honest node. Thus, strong anonymity cannot be achieved.

3) \(\ell = c = K - 1\):

Now we have, \(\left( \frac{\gamma}{\ell} \right) = \frac{c}{K - 1} = \frac{c}{c/\ell}\). But \(K \in \text{poly}(\eta)\), and \(c\) and \(K\) only have integer values. Hence \((c - \ell)/(K - c)\) is non-negligible, and hence \(\left( \frac{\gamma}{\ell} \right)\) is also non-negligible. Even though \(c\) has a small value, \(\ell\) is also small. Hence, it is unlikely that the challenge message will mix with a message from \(u_{1-2}\) at some honest node. Thus, strong anonymity cannot be achieved, despite the necessary constraints being satisfied.

#### Theorem 5.

For user distribution \(U_B\) with \(K \in \text{poly}(\eta)\), \(c \in \Omega(1)\), \(K > c \geq 1\), \(\ell < \eta\) and \(\beta N \geq 1\), no protocol \(\Pi \in M\) can achieve strong anonymity if \(2\ell \beta < 1 - \epsilon(\eta)\), where \(\epsilon(\eta) = 1/\eta^4\) for a positive constant \(d\).

**Proof.** When \(c < \ell\): \(\delta \geq 1 - \left( \frac{1}{\ell} \right) f_{\beta}(\ell)\) and \(f_{\beta}(\ell)\) to become overwhelming. From Theorem 2 and Theorem 1, we know that \(2\ell \beta > 1 - \text{neg}(\eta)\) is a necessary condition for \(f_{\beta}(\ell)\) to become overwhelming. Now, we are left with the factor \(1 - \left( \frac{1}{\ell} \right) f_{\beta}(\ell)\). This can become overwhelming iff \(\left( \frac{\gamma}{\ell} \right) f_{\beta}(\ell)\) becomes negligible. We know that \(K > c \geq \ell\) and \(K \in \text{poly}(\eta)\). Hence, for some constant \(x\),

\[
\frac{c - \ell}{K - \ell} > \frac{1}{\eta^x} \iff \left( \frac{c - \ell}{K - \ell} \right)^x > \left( \frac{1}{\eta^x} \right)^x
\]

\[
\implies \left( \frac{c - \ell}{K - \ell} \right)^x > \frac{1}{\eta^x}
\]

For any \(\ell \in \Omega(1)\), \((1/\eta^x)^x\) is non-negligible.
And we already know that \( [1 - 1/\binom{N}{2}] \) can never be overwhelming. Thus, the only way \( \delta \) can become negligible is if \( f_\delta(\ell - c) \) becomes overwhelming. Note that, if \( a + b \leq 1 \) and \( c < 1 \), the only way \( ac + b = 1 \) is possible if \( b = 1 \).

Now we can follow exactly the same procedure as in the proof of Theorem 2 to say: \( f_\delta(\ell - c) \) can not become overwhelming if \( 2(\ell - c)\beta < 1 - \epsilon(\eta) \).

The analysis in this case is exactly same as Section V-B, except that here we need to consider the slice of \((\ell - c)\) rounds instead of \( \ell \) rounds.

It is worth repeating here, all the constraints we have derived in Section V and Section VI are necessary for anonymity, but they are not sufficient conditions for anonymity.

VII. UNSYNCHRONIZED USERS WITH NON-COMPROMISING ADVERSARIES

In this and the subsequent section we use an unsynchronised user distribution \( U_p \): In each round, independent of other users and other rounds, each client tosses a biased coin with success probability \( p \in (0, 1) \). On a success the client sends a message in that round, otherwise it does not send a message. Consequently, the number of messages per round follows Binomial distribution \( Binom(N, p) \) if the number of users \( N \) is large and \( p \) sufficiently small, the resulting binomial distribution reduces to a Poisson distribution, which is a close approximation of real-life traffic patterns.

For a protocol with bandwidth overhead \( \beta \), we distinguish between the actual probability that users want to send messages \( p' \) and the value for \( p \) that we use in our analysis, i.e., we set \( p = p' + \beta \). In this unsynchronised scenario the bandwidth of genuine messages contributes to the anonymity bound.

As in Section V, in this section we consider a non-compromising adversary.

A. Lower Bound on Adversarial Advantage

**Theorem 6.** For user distribution \( U_p \), no protocol \( \Pi \in M \) can provide \( \delta \)-sender anonymity, for any \( \delta < 1 - (\ell/2 + f_\ell(\ell)) \), where \( f_\ell(x) = \min\{1/2, 1 - (1 - p)^x\} \) for a positive integer \( x \).

**Proof.** Similar to Section V, we calculate a bound on the probability that \( \Pi_{\text{ideal}} \) satisfies Invariant 1, and that bound is valid against any other protocol in our model. Since we consider a non-compromising adversary, \( \Pr[\text{Invariant 1 is True}] = \Pr[T \text{ is not empty}] \), where \( T \) is defined as in Invariant 1.

Let us consider the random variables \( X^{(1)}, X^{(2)}, \ldots, X^{(N)} \), where \( X^{(i)} \) denotes the event of the \( i^{\text{th}} \) user sending her own message within a given interval of \( \ell \) rounds \([a, b]\), with \((b - a) = \ell \). All \( X^{(i)} \)'s are mutually independent and we have,

\[
X^{(i)} = \begin{cases} 
0 & \text{with probability } (1 - p)^\ell \\
1 & \text{with probability } (1 - (1 - p)^\ell).
\end{cases}
\]

Next, let \( X = \sum_{i=1}^{N} X^{(i)} \) be a random variable representing the number of users that send messages in an interval of \( \ell \) rounds. We calculate for the expected value \( \mathbb{E}[X] \) of \( X \),

\[
\mathbb{E}[X] = \mathbb{E}[\sum_{i=1}^{N} X^{(i)}] = \sum_{i=1}^{N} \mathbb{E}[X^{(i)}] = N(1 - (1 - p)^\ell) = \mu.
\]

Using the Chernoff Bound on the random variable \( X \) we derive \( \Pr[X - \mu > N\eta] \leq \exp(-2a^2N) \), where \( a = \frac{\mu}{N} \) is the value for \( X \) and the event that \( T \) is non-empty by \( Y \) and since all users are acting independently from each other we get for \( j \in \{0, \ldots, N\} \), \( \Pr[Y; X = j] = 1 - \Pr[|Y; X = j| = \frac{j}{N}] \).

For \( 2\mu \leq N \), we have the following,

\[
\Pr[Y] = \Pr[X \geq 2\mu] \times \Pr[Y; X \geq 2\mu] \times \Pr[Y; X < 2\mu] \\
\leq \Pr[X \geq 2\mu] \times \Pr[Y; X = N] + \Pr[X < 2\mu] \times \Pr[Y; X = 2\mu] \\
= \mathbb{E} \times \Pr[Y; X = N] + (1 - \mathbb{E}) \times \Pr[Y; X = 2\mu] \\
= \mathbb{E} \times \frac{N}{N} + (1 - \mathbb{E}) \times \frac{2\mu}{N} = 1 - (1 - \mathbb{E})(1 - 2f_\ell(\ell)) \cdot
\]

If \( 2\mu > N \), we get with \( f_\ell(\ell) = \min\{\frac{1}{2}, 1 - (1 - p)^\ell\} \), \( \Pr[Y] \leq E + (1 - E)(1 < 1 - (1 - (2f_\ell(\ell))) \).

Thus, \( \delta > 1 - \Pr[Y] \geq (1 - E)(1 - 2f_\ell(\ell)) \). We now use Markov’s Inequality on \( X \) and derive \( E = \Pr[X = 2\mu] \leq \frac{1}{2} \), which means, \( \delta \geq \frac{1}{2} (1 - 2f_\ell(\ell)) \geq \frac{1}{2} - f_\ell(\ell) \).

Note that in the proof of Theorem 6, in case \( p \) is a constant and \( N \) is a very high value, then \( E \) goes towards zero and instead of using Markov’s inequality, we can derive \( \delta \geq 1 - 2f_\ell(\ell) \).

B. Impossibility for Strong Anonymity

**Theorem 7.** For user distribution \( U_p \) and \( p > 0 \), no protocol \( \Pi \in M \) can achieve strong anonymity if \( 2\ell p < 1 - \epsilon(\eta) \), where \( \epsilon(\eta) = 1/\eta^4 \) for a positive constant \( d \).

**Proof.** We know \( 0 \leq E \leq 1/2 \). When \( 2\mu \leq N \),

\[
\delta \geq (1 - E)(1 - 2f_\ell(\ell)) \geq \frac{1}{2} \cdot \left( 2(1 - p)^\ell - 1 \right) \\
\geq \frac{1}{2} \cdot (2(1 - \ell p)(1 - 1) = 1/2(1 - 2\ell p).
\]

Thus, if \( 2\ell p < 1 - \epsilon(\eta) \),

\[
2\ell p < 1 - \epsilon(\eta) \iff 1 - 2\ell p > \epsilon(\eta) \\
\iff \delta > 1/2 \times \epsilon(\eta) \iff \delta \text{ non-negligible}.
\]

Thus, when \( 2\mu \leq N \), a necessary condition for \( \delta \) to become negligible is \( 2\ell p > 1 - \eta \).

When \( 2\mu > N \), using \( \mu = N(1 - (1 - p)^\ell) \) we get:

\[
2N(1 - (1 - p)^\ell) > N \iff (1 - p)^\ell < 1/2 \\
\iff 1 - \ell p < 1/2 \iff 2\ell p > 1.
\]

Similar to the constraints in Section V and Section VI, this is also a necessary constraint for anonymity, not a sufficient condition. There can exist \( \ell \) and \( p \) such that \( 2\ell p > 1 - \eta \), but still no protocol can achieve strong anonymity.

**Interesting Cases.** Now we are going to discuss a few interesting cases for different values of \( \ell, p, \) and \( N \).

1) \( p = \frac{1}{\eta^2} \), \( \ell = \eta \): Here, \( f_\ell(\ell) = 1 - (1 - p)^\ell > 1 - 1/e > 2/3 \).

Hence, \( \delta \geq \frac{1}{2} - f_\ell(\ell) \). Since \( \ell p = 1 \), in \( \ell \) rounds the protocol has 1 message per user on average. So, the protocol
has a high chance of winning, but depending on the specific instance of the user distribution. Whereas in Section V-B, we saw that, for a similar bandwidth and latency overhead, protocols could win with all instances of the synchronized user distribution.

2) \( p = \frac{1}{2\eta}, \ell = \eta; \) even for \( \eta > 2 \), \( f_p(\ell) = 1 - (1 - p)^\ell < 0.45 \). Hence, \( \delta \geq \frac{1}{2} - f_p(\ell) > 0.05 \). Even though 2\ellp = 1, strong anonymity can not be achieved in this case. In an expected scenario, in a slice of \( \ell \) rounds only \( p\ell = \frac{1}{2} \) portion of the total users sends messages, and hence there is a significant chance that \( u_{1-b} \) is in the other half. Note that this is different from the scenario with synchronized users where protocols could achieve strong anonymity in this case (c.f. Section V-B).

3) \( p = \frac{1}{2}, \ell = 1: \) We have, \( f_p(\ell) = 1 - (1 - p)^\ell = \frac{1}{2} \). Hence, \( \delta \geq 0 \). Although we have 2\ellp = 1, because of low \( \ell = (1) \), \( u_{1-b} \) does not send a message with high probability (= \( \frac{1}{2} \)). This case again highlights that the requirement 2\ellp \geq 1 - \epsilon(\eta) is not necessarily sufficient: As in Section V-B, protocols can not achieve strong anonymity in such a situation.

4) \( p = \frac{1}{2}, \ell = 3: \) Here, \( f_p(\ell) = 1 - (1 - p)^\ell = 1 - (\frac{1}{2})^3 < 0.29 \), and \( \delta \geq \frac{1}{2} - f_p(\ell) > 0.21 \); because of low values of both \( p \) and \( \ell \) only a few users send messages within the interval of \( \ell \) rounds, and hence the protocol has a small chance to win. As in Section V-B, protocols can not achieve strong anonymity in this case, since the necessary constraints are not satisfied.

VIII. UNSYNCHRONIZED USERS WITH PARTIALLY COMPROMISING ADVERSARIES

Finally, we consider partially compromising adversaries that can compromise a set of \( c \) parties \( P_c \subset P \) for the user distribution \( U_P \) defined in Section VII.

A. LOWER BOUND ON ADVERSARIAL ADVANTAGE

**Theorem 8.** For user distribution \( U_P \), no protocol \( \Pi \in M \) can provide \( \delta \)-sender anonymity, for any

\[
\delta < \left\{ \begin{array}{ll}
1 - \frac{1}{(\ell+1)} f_p(\ell) & c \geq \ell \\
\left(1 - \frac{1}{(\ell+1)}\right) \left[ \frac{1}{2} + f_p(\ell) \right] & c < \ell
\end{array} \right.
\]

where \( f_p(x) = \min(\frac{1}{2}, 1 - (1 - p)^x) \) for a positive integer \( x \).

We derive the bound in Appendix B by combining the techniques presented in Section VI and Section VII. Since the proof does not introduce novel techniques, we omit it and instead refer the interested reader to Appendix B for the proof.

B. IMPOSSIBILITY FOR STRONG ANONYMITY

To analyze the negligibility condition of \( \delta \) in this scenario, we heavily borrow the analyses that we already have conducted in Section VII-B and Section VI-B. We are going to analyze this scenario in two parts:

**Case c \geq \ell:** We have, \( \delta \geq 1 - \frac{1}{(\ell+1)} \left[ \frac{1}{2} + f_p(\ell) \right] \).

To make \( \delta \) negligible, both the factors \( \frac{1}{(\ell+1)} \) and \( \frac{1}{2} + f_p(\ell) \) have to become overwhelming. From Theorem 4, we know that we need \( \ell \in \omega(1) \) to make \( \frac{1}{(\ell+1)} \) overwhelming. This is a necessary condition, but not sufficient.

For a detailed discussion, we refer to Section VI-B. From Section VII-B we know that the necessary condition for \( \frac{1}{2} + f_p(\ell) \) to be overwhelming is \( 2\ellp > 1 - \epsilon(\eta) \). Hence, both conditions are necessary to achieve strong anonymity.

**Case c < \ell:** We have,

\[
\delta \geq \left(1 - \left[\frac{1}{2} + f_p(\ell - c)\right]\right) \left(1 - \frac{1}{(\ell+1)}\right) \left[\frac{1}{2} + f_p(\ell)\right].
\]

In the above expression, we can see two factors:

(i) \( F_1 = (1 - \left[\frac{1}{2} + f_p(\ell - c)\right]) \), (ii) \( F_2 = (1 - \frac{1}{(\ell+1)}\left[\frac{1}{2} + f_p(\ell)\right]) \).

To make \( \delta \) negligible, it suffices that \( F_1 \) or \( F_2 \) become negligible. Unlike Section VI, here \( f_p(\ell - c) \) and \( f_p(\ell) \) are independent, which allows us to analyze \( F_1 \) and \( F_2 \) independently. First, \( F_1 \) is similar to the \( \delta \)-bound in Section VII, except that we consider \( f_p(\ell - c) \) instead of \( f_p(\ell) \). Hence, the analysis of \( F_1 \) is analogous to Section VII-B. Second, \( F_2 \) is negligible if both \( \left[\frac{1}{2} + f_p(\ell)\right] \) and \( \left[\frac{1}{2} + f_p(\ell)\right] \) are overwhelming. From Section VI-B we know that \( \left[\frac{1}{2} + f_p(\ell)\right] \) can not be overwhelming for a constant \( c \). Moreover, \( f_p(\ell) \) can be analyzed exactly as \( f_p(\ell) \) in Section VII-B.

IX. RECIPIENT ANONYMITY

For recipient anonymity, we analyze the adversary’s success in determining the recipient of a particular message sent by a user. Moreover, the adversary is naturally not informed about the delivery of the challenge message by a recipient, but of the sending of the challenge message by a user. We derive impossibility results for recipient anonymity analogous to our results for sender anonymity via the same strategy we employed in the previous sections. In this case, since we are considering recipient anonymity, we assume only one sender in \( S \), and \( N' \) users in \( R \). Moreover, instead of ignoring all internally generated messages in Claim 2 we ignore all internally terminating messages. Note that this gives \( \beta \) a slightly different flavor.

For a detailed recipient-anonymity analysis, we refer the readers to Appendix C.

**Synchronized Users.** We slightly tweak the user distribution to suit the definition of recipient anonymity. We assume that all the input messages come within \( N' \) rounds, exactly one message per round, following a random permutation that assigns one round to each recipient. In a given round, the sender sends a message to the assigned recipient. Then, the protocol decides when to deliver the message to the recipient, but not delaying more than \( \ell \) rounds. Let \( f_p^{\beta}(x) = \min(1, \frac{1}{N'}(x + \ell) + (x + \ell)\beta) \)

Then we get that no protocol \( \Pi \in M \) can provide \( \delta \)-recipient anonymity in the following cases:

- Without compromisation: \( \delta < 1 - f_p^{\beta}(\ell) \).
- For adversaries that compromise up to \( c \) parties:
  - if \( c \geq \ell \): \( \delta < 1 - \frac{1}{(\ell+1)} f_p^{\beta}(\ell) \).
  - if \( c < \ell \): \( \delta < 1 - \frac{1}{(\ell+1)} f_p^{\beta}(\ell - c) \).

Moreover, no protocol in \( M \) with \( K \in \text{poly}(\eta) \) can achieve strong recipient anonymity when \( \ell < N' \) and \( \beta N' \geq 1 \) in the following cases, where \( \epsilon(\eta) \) is a non-negligible function.

- Without compromisation: if \( A \beta < 1 - \epsilon(\eta) \).
- For adversaries that compromise up to \( c \) parties:
- if $K > c \geq \ell$: $4\ell\beta < 1 - \epsilon(\eta) \text{ OR } \ell \in O(1)$.
- if $K > \ell > c$: $4(\ell - c)\beta < 1 - \epsilon(\eta)$.

**Unsynchronized Users.** Similar to the previous case, here also we borrow the definition of user distribution from Section VII, with minor modifications. The biased coins are now associated with recipients instead of senders — in each round the sender sends a message for a recipient, with probability $p$. Let $f^p_\beta(x) = \min(1/2, 1 - (1 - p)^{\beta x})$. Then we get that no protocol $\Pi \in M$ can provide $\delta$-recipient anonymity in the following cases:
- Without compromisation: $\delta < 1 - (1/2 + f^\Pi_{\beta}(\ell))$.
- For adversaries that compromise up to $c$ parties:
  - If $c \geq \ell$: $\delta < \left(1 - 1/2 + f^\Pi_{\beta}(\ell - c)\right) \times \left(1 - 1/(f^\Pi_{\beta}(c))\right)$.
  - If $c < \ell$: $\delta < \left(1 - 1/2 + f^\Pi_{\beta}(\ell - c)\right) \times \left(1 - 1/(f^\Pi_{\beta}(c))\right)$.

Moreover, for $p > 0$, no protocol can achieve strong recipient anonymity if $2\ell p < 1 - \epsilon(\eta)$, where $\epsilon(\eta)$ is a non-negligible function.

**X. Implications**

To put our result into perspective, we discuss whether our trilemma excludes strong anonymity for a few AC protocols from the literature. More precisely, this section exemplarily applies the results from Theorem 2 and Theorem 7, i.e., with synchronized and unsynchronized user distributions and a global network-level, non-compromising adversary. We use both results since for some AC protocols (e.g., DC-nets [15]) the synchronized user distribution is more accurate and for other protocols (e.g., Tor [10]) the unsynchronized user distribution is more accurate. Our constraints mark an area on a 2D graph (see Figure 7) with latency overhead (x-axis) versus bandwidth overhead (y-axis) where strong anonymity is impossible. As the latency of some AC protocols depends on system parameters and we want to place the protocols in a 2D graph, we carefully choose system parameters and make a few simplifying assumptions, which are subsequently described.

This section is solely intended to put our impossibility result into perspective by viewing how we estimated the bandwidth $\beta$ and latency $\ell$ bounds in the sense of this work. It is not meant and not qualified to be a performance and scalability comparison of the discussed AC protocols, which would have to take many other dimensions into account, e.g., the communication and computation complexity of the servers and the receivers, the computation complexity of the senders and the different kinds of functionalities that are offered by the different AC protocols (e.g., group communication vs. internet-like visitor-webpage communication). Table I summarizes bounds on the bandwidth $\beta$ and latency overhead $\ell$ (in the sense of this work).

Table: Latency vs. bandwidth vs. strong anonymity of AC protocols, with the number of protocol-nodes $K$, number of clients $N$, and message-threshold $T$, expected latency $\ell^*$ per node, dummy-message rate $\beta$;

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Latency</th>
<th>Bandwidth</th>
<th>Strong Anonymity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tor [10]</td>
<td>$\theta(1)$</td>
<td>$\theta(1/N)$</td>
<td>impossible</td>
</tr>
<tr>
<td>Hornet [49]</td>
<td>$\theta(1)$</td>
<td>$\theta(1/N)$</td>
<td>impossible</td>
</tr>
<tr>
<td>Herd [25]</td>
<td>$\theta(1)$</td>
<td>$\theta(N/N)$</td>
<td>possible</td>
</tr>
<tr>
<td>Riposte [50]</td>
<td>$\theta(N)$</td>
<td>$\theta(N/N)$</td>
<td>possible</td>
</tr>
<tr>
<td>Vuvuzela [20]</td>
<td>$\theta(K)$</td>
<td>$\theta(N/N)$</td>
<td>possible</td>
</tr>
<tr>
<td>Riffle [21]</td>
<td>$\theta(K)$</td>
<td>$\theta(N/N)$</td>
<td>possible</td>
</tr>
<tr>
<td>Threshold mix [14]</td>
<td>$\theta(T\ell)$</td>
<td>$\theta(1/N)$</td>
<td>impossible$^*$</td>
</tr>
<tr>
<td>Loopix [24]</td>
<td>$\theta(\sqrt{\ell})$</td>
<td>$\theta(\beta)$</td>
<td>possible</td>
</tr>
<tr>
<td>DC-Net [15,18]</td>
<td>$\theta(1)$</td>
<td>$\theta(N/N)$</td>
<td>possible</td>
</tr>
<tr>
<td>Dissent-AT [22]</td>
<td>$\theta(1)$</td>
<td>$\theta(N/N)$</td>
<td>possible</td>
</tr>
<tr>
<td>DiceMix [16]</td>
<td>$\theta(1)$</td>
<td>$\theta(N/N)$</td>
<td>possible</td>
</tr>
</tbody>
</table>

* if $T$ in $o(poly(\eta))$

Including the bandwidth overhead they introduce over the plaintext. We assume an upper bound on the latency of the protocol and are oblivious to server-side noise (see Claim 2). Moreover, recall that we are only interested in the question whether our trilemma excludes strong anonymity for the ten AC protocols from the literature; hence, we consider the upper bound on the latency and bandwidth overhead for deterministic latency. For randomized latency, such as Loopix [24], we list for simplicity the expected delay as the latency bound.

**Low-latency protocols** such as Tor [10], Hornet [49], and Herd [25] are low-latency AC protocols, i.e., they immediately forward messages. While Tor and Hornet do not produce asymptotically more than a constant amount of both bandwidth overhead and latency overhead and thus cannot provide strong anonymity, Herd produces dummy traffic linearly proportional to the number of users (bandwidth overhead $\beta \in \theta(N/N)$), thus the trilemma does not exclude strong anonymity for Herd.

**Riposte** [50] uses secure multiparty computation and a variant of PIR to implement an anonymous bulletin board. Riposte operates in epochs and for each epoch the set of users is public. Hence, Riposte is expected to be run with long epochs to maximize the number of users that participate in an epoch, which leads us to estimating the latency overhead to be $\ell \in \theta(N)$. To counter traffic analysis attacks, Riposte clients send constant dummy traffic, resulting in a bandwidth overhead of $\beta \in \theta(N/N)$. Thus, the trilemma does not exclude strong anonymity for Riposte.

**Vuvuzela** [20] is a mix-net that is tailored towards messengers. Clients communicate by depositing their encrypted messages in one of the mix net nodes. To achieve strong resistance against compromised servers, Vuvuzela takes a path through all servers, resulting in a latency overhead of $\ell \in \theta(K)$ (for $K$ servers). Additionally, Vuvuzela utilizes constant traffic, leading to a bandwidth overhead of $\beta \in \theta(N/N)$, and has the potential for strong anonymity.

**Riffle** [21] uses a verifiable mix-net, however not only for messenger communication but also for normal client-server web traffic. Just as Vuvuzela, Riffle also chooses paths that traverse all K servers, leading to $\ell \in \theta(K)$ and if we assume
to every other party. As our bandwidth overhead only counts dummy-message rates, it does not capture the broadcast, thus $\beta \in \theta(N/N)$. DC-nets use a combination operation (a simple XOR in Chaum’s original paper) that causes dummy messages to cancel out. Then, all parties output the resulting bitstring. If only one real message is sent, the bitstring equals this message. As Theorem 7 assumes a synchronized user distribution, in each round only one party sends a message, thus our model treats $\ell$ as $\ell \in \theta(1)$.

The Dissent-AT [22] scheme (the AnyTrust-variant of Dissent) improves on the performance of DC-nets by relying on dedicated servers. Instead of sending in each round fake or real ciphertexts to every other client, clients in Dissent-AT send these messages to at least one of these dedicated servers. These servers then perform a DC-net communication round. Abstracting from an initial set-up phase and only counting the client-messages, Dissent-AT has $\beta \in \theta(N/N)$ for the clients (assuming that each client communicates to one server), and $\ell \in \theta(1)$.

Dicemix [16] is a peer-to-peer AC protocol that is based on the DC-net approach. While Dicemix includes a self-healing mechanism that leads to $4 + 2f$ communication rounds for one message if $f$ peers are malicious, this mechanism does not kick in if all peers are honest, leading to only 4 communication rounds. The authors additionally had the insight that a trusted party, i.e., a bulleting board, can be used for the broadcast. This party can even be malicious in which case the bulleting board can stop the protocol but not deanonymize the parties. This bulleting boards keeps the latency at 4, which is in $\theta(1)$. As every party sends a message in every round, $\beta \in \theta(N/N)$.

Fig. 7. Asymptotic latency overhead ($\ell$) and bandwidth overhead ($\beta$) together with the “area of impossibility” where $2\ell\beta \leq 1 - \frac{1}{\log(\eta)}$. We portray protocols as dots depending on their choices for $\ell$ and $\beta$. Technically, if we use Theorem 7, we $\beta$ is replaced by $p = \beta + p'$, where $p'$ is the rate at which users send messages. This graph assumes $N$ is ca. $\log(\eta)$, the number of nodes $K$ is $\log(\eta)$. The threshold for Threshold Mix $T = 1$ and for Threshold Mix$_{\sec}$ $T = N = \log(\eta)$. In the graph, both the axes are approximately in logarithmic scale. (For a more accurate visual representation we refer the readers to Appendix D and [33].)

$K \in \theta(\log(\eta))$, we get $\ell \in \theta(\log(\eta))$. We assume that the clients send dummy traffic up to a constant rate (depending on the user’s sending rate $p'$), so we have $\beta \in \theta(N/N)$ and the potential for strong anonymity.

In a threshold mix net, each of the $K$ mix servers waits until it received up to a threshold $T$ many messages before relaying the messages to the next mix, resulting in $\ell \in \theta(T \times K)$. Threshold mixes [14] do not provide strong anonymity unless their threshold $T$ is of the order of the number of users $N$. As such a large threshold are impractical for a large number of users, we judge it impossible to achieve strong anonymity for practical deployments of Threshold mixes.

Loopix [24] is a mix net that combines exponentially distributed delays at each mix-node and dummy messages from each user. Ignoring so-called loop messages (meant to counter active attacks), Loopix naturally enforces our unsynchronised user distribution: the rate at which Loopix clients send messages is the sum of a dummy-message rate ($\beta$) and a payload message rate ($p'$), which are system parameters. We assume that the path lengths in Loopix’ stratified topology is $\sqrt{K}$ with the number of nodes $K \in \theta(\log(\eta))$. If $\beta + p' \geq 1/\sqrt{\eta}$, and if every hop introduces an expected delay of $\ell' \geq \sqrt{\eta}$, the expected latency overhead is $\ell = \sqrt{K} \times \ell'$, in particular $\ell \in \theta(\sqrt{\eta})$. We get $(p' + \beta)\ell = \frac{1}{\sqrt{\eta}} \times \sqrt{\eta} = 1$ and the trilemma does not exclude strong anonymity for Loopix, which grants the protocol an interesting spot in our figure.

In AC protocols based on DC-nets [15], [18] every party broadcasts either a dummy or real message in every round to other party. As our bandwidth overhead only counts dummy-message rates, it does not capture the broadcast, thus $\beta \in \theta(N/N)$. DC-nets use a combination operation (a simple XOR in Chaum’s original paper) that causes dummy messages to cancel out. Then, all parties output the resulting bitstring. If only one real message is sent, the bitstring equals this message. As Theorem 7 assumes a synchronized user distribution, in each round only one party sends a message, thus our model treats $\ell$ as $\ell \in \theta(1)$.

The Dissent-AT [22] scheme (the AnyTrust-variant of Dissent) improves on the performance of DC-nets by relying on dedicated servers. Instead of sending in each round fake or real ciphertexts to every other client, clients in Dissent-AT send these messages to at least one of these dedicated servers. These servers then perform a DC-net communication round. Abstracting from an initial set-up phase and only counting the client-messages, Dissent-AT has $\beta \in \theta(N/N)$ for the clients (assuming that each client communicates to one server), and $\ell \in \theta(1)$.

Dicemix [16] is a peer-to-peer AC protocol that is based on the DC-net approach. While Dicemix includes a self-healing mechanism that leads to $4 + 2f$ communication rounds for one message if $f$ peers are malicious, this mechanism does not kick in if all peers are honest, leading to only 4 communication rounds. The authors additionally had the insight that a trusted party, i.e., a bulleting board, can be used for the broadcast. This party can even be malicious in which case the bulleting board can stop the protocol but not deanonymize the parties. This bulleting boards keeps the latency at 4, which is in $\theta(1)$. As every party sends a message in every round, $\beta \in \theta(N/N)$.

XI. CONCLUSION AND FUTURE WORK

This paper proves the anonymity trilemma: strong anonymity, low bandwidth, low latency—choose two! We derive necessary constraints for sender anonymity and recipient anonymity, and thereby presents necessary constraints that are crucial for understanding bi-directional anonymous communication: sender anonymity for hiding the sender and recipient anonymity for hiding the recipient of a message. To put our result in perspective, we evaluate how ten relevant AC protocols from the literature cope with the trilemma.

For future work, we plan to extend the work in four natural directions: (i) derive tighter bounds by using more sophisticated attackers, (ii) derive bounds for other anonymity notions (e.g., unlinkability and relationship anonymity), (iii) extend the protocol mode with a notion of a throughput limitation, (iv) relax the requirement that messages are sent with certainty and allow for unreliable channels. For example, for the first direction, we plan to take the same steps as outlined in Section II-B, i.e., to formulate an invariant, to construct a protocol optimal w.r.t. this invariant, and then to compute the advantage of the more sophisticated attacker against this protocol.
ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments. This work has been partially supported by the Zurich Information Security Center (ZISC), the European Commission through H2020-DS-2014-653497 PANORAMIX, the EPSRC Grant EP/M013-286/1, and the National Science Foundation (NSF) under grant CNS-1719196.

REFERENCES

APPENDIX A

Protocol Model Revisited

A. Validity of the Protocol Model (Contd.)

Lemma 2. Let $\Pi$ be a protocol $\in M$ with $K$ parties with parameters $\beta$ and $\ell$. Then: 1) Messages are delivered within $\ell$ steps. 2) The protocol adds (for the unsynchronised case on average) a maximum of $\beta$ noise messages per user per round. 3) Whenever a party in $S \cup P$ sends a message to another party in $P \cup R$, the adversary learns that and in which round this happens. 4) For every message that leaves the network (received by $R$), the adversary additionally learns whether the message is the target message. 5) For every compromised party, the adversary learns the mapping between the input messages and the output messages.

Proof. Let $\Pi$ be a protocol $\in M$ with $K$ parties with parameters $\beta$ and $\ell$. We analyze the lemma part by part.

1) Messages are delivered within $\ell$ steps.
2) The protocol adds (for the unsynchronised case on average) a maximum of $\beta$ noise messages per user per round.
3) The adversary learns that and in which round a party in $S \cup P$ sends a message to another party in $P \cup R$.
4) For every message that leaves the network (received by $R$), the adversary additionally learns whether this message is the target message.
5) For every compromised party, the adversary learns the mapping between the input messages and the output messages.

Part (2) of the Lemma holds, since we restrict the user distributions accordingly and since the none of the transitions in the Petri net can create more tokens within the network than it consumes from its input place.

We show the part (1) of the lemma via structural induction over fired transitions of the Petri net. We additionally add to the induction invariant that all tokens that are not in $S$ have a timestamp for their next transition of $ts = 1$ and a remaining time of $t_r > 0$ and there are at least $t_r$ rounds left in which the token can be delivered.

Induction base: The protocol is initialized and no transitions have happened. Thus, no messages have been sent so far, i.e., there is no message that has not been delivered within $\ell$ steps. The only transition that can fire is $TS$ and for $\ell > 0$, the message introduced into the network in this way does not need to be delivered already ($0 < t_r = \ell$). Moreover, $TS$ sets the timestamp of this message token to $ts = 1$.

Induction step: Let $tr$ be any execution trace s.t. the induction invariant is satisfied and let $t$ be an arbitrary possible transition that extends $tr$ to $tr : t$.

We distinguish two cases for $t$: In case $t$ is $TS$, it consumes a token from $P_S$ and puts this token into a place $P_i$ and, by definition we have $t_r > 0$ and $ts = 1$. Otherwise, the transition is $TP_i$ for some $i$ and consumes a token from $P_i$ accordingly. By the induction invariant, the token has $t_r > 0$. If this token has $t_r = t - 1 = 0$, the transition delivers the token to $R$. Otherwise, $t$ decreases $t_r$ by one (thus fulfilling the condition that there are at least $t_r$ rounds left in which the token can be delivered) and sets $ts = 1$. Since every token in any place $P_i$ needs to be consumed in every round, the protocol delivers every message in at most $\ell$ steps.

Other parts of the lemma: By definition of our Petri net, whenever a transition fires, an element $(t, r)$ is placed into Tokens, containing the public fields of $t$, such as $t.prev$ and $t.next$, as well as the current round number $r$, which fulfills part (3). Moreover, whenever the transition places the token in $R$, the adversary can additionally see the field $t.msg$ and no transition can change the field $msg$, which allows the adversary to effectively tag and recognize the challenge message and thus fulfills part (4). Finally, if any party $P_i$ is compromised, $P_i$ does not modify the unique (and otherwise freshly sampled) field $t.ID$, which allows the adversary to map incoming and outgoing messages.

Since the transitions discussed here are the only way for messages to be sent to a recipient, the model correctly enforces the conditions from the lemma.

B. Expressing Protocols in the Petri net model

Modeling DC net. Here we show how to model an actual DC net type protocol using our Petri net model $M$ as defined in Section IV. Specifically we pick up the short DC net protocol proposed by Golle and Juels [18], and present $M_{\text{DC}}$ which models the aforementioned protocol.

We model a DC net protocol with $N$ participants, where $S = P_i, |S| = |P| = N$. We denote the parties with $P_1, \ldots, P_N$. The protocol can be denoted by $M_{\text{DC}} = \{\text{paramgen, keydist, post, verify, extract}\}$ as described below.

- paramgen: In $\text{prot}_{\text{DC}}$, paramgen is executed by a trusted entity and the output is published. Since we are mainly interested in the anonymity game, we consider that paramgen step is executed by our honest challenger and happens outside the protocol run, and the output is globally known (to all the transitions $T_{P_i}$).

- keydist: using the output of paramgen, this step yields for each party $P_i$, a private key $x_i$ and a corresponding public key $y_i$. In $\text{prot}_{\text{DC}}$, the above key generation part is done by a trusted entity, and hence we consider that it is done by our honest challenger and for each party $P_i$ the public-private keypair $x_i, y_i$ is already known to the corresponding transition

\[8\text{Since we are mainly interested in the anonymity property, we don’t need to model the part of the protocol where the protocol parties reconstructs the keys in case of a failure. But it is easy to extend $M_{\text{DC}}$ to include that step by adding one more round to the current model.}\]
function $T_P$. As part of protocol each party $P_i$ publishes its public key $y_i$. Additionally, each party $P_i$ receives from $P_i$ a share of private key $x_{i,j}$ and a share of public key $y_{i,j}$, where the keys are shared in a $(k, N)$ threshold manner for a parameter $k \leq N$.

- **post**: Each player $P_i$ generates a vector of random pads $W_i = \{W_i(1), W_i(2), \ldots, W_i(N)\}$ using $x_i$. $\Pi_{DC}$ does not handle collisions, instead assumes that the players decide their positions by a consensus protocol. Similarly our model assumes that each party $P_i$ knows its position, and assume the position is $q_i$ (but not known to the adversary). Then each player $P_i$ computes the vector $V_i$ such that $V_i(w) = W_i(w)$ for all $w \neq i$ and $V_i(w) = W_i(w) \oplus m_i$ for $w = q_i$, where $m_i$ is the message of $P_i$. Also, each player $P_i$ computes $\sigma_i = \{\sigma_i(1), \sigma_i(2), \ldots, \sigma_i(N)\}$, where $\sigma_i$ includes the identity of player $P_i$ and a proof of valid formatting of $V_i$. Then $P_i$ publishes both the vectors $V_i$ and $\sigma_i$. Our model assumes the pair $(V_i(w), \sigma_i(w))$ for each position $w$ as a single message, where $V_i(w)$ is a message content and $\sigma_i(w)$ becomes a part of meta field. For each position $w$ player $P_i$ generates one such message, and publishes the message to all other players.

- **verify and extract** are local computations after a party $P_i$ receives messages from all other parties.

Although the protocol model assumes that the adversary can not read the contents of any message, here we shall model $\Pi_{DC}$ along with its cryptographic primitives to demonstrate the expressiveness of our model. Alternatively, to get rid of all the cryptographic primitives, the parties can send a dummy message ($= 0$) whenever $V_i(w) = W_i(w)$, and the actual message $m_i$ whenever $V_i(w) \neq W_i(w)$.

As per our anonymity definition in Section III, we assume that up to $(N-2)$ users can be compromised, which necessarily makes up to $(N-2)$ protocol parties compromised. The adversary chooses two challenge users, and one of them sends the challenge message depending on the challenge bit $b$. All other $(N-1)$ users send dummy messages.

In $\Pi_{DC}$ we model $\Pi_{DC}$ as a two round protocol. The challenger sets the initial configuration of the Petri net with the messages to be sent by each party. In the first round, each party $P_i$ sends two kinds messages: (1) the public key message $y_i$ and (2) sends share of the public-private keypair $(x_{i,j}, y_{i,j})$ to $P_j$ for all $j \neq i$. Here, one party can publish a message to $(N-1)$ other parties by sending $(N-1)$ separate messages. In the second round, each party $P_i$ publishes $N$ messages: one message for each position, only one of them contains his own message. After second round, every party receives messages from every other party, and then do local computations to verify and extract the original messages.

For $\Pi_{DC}$, we do not actually need a separate recipient $R$ in $\Pi_{DC}$, if we make $R = P$. But, to be consistent with $M$, in $\Pi_{DC}$ we keep a separate recipient. In the second round whenever a party $P_i$ publishes a message, $P_i$ also sends a copy to $R$. This easily models the fact that the adversary knows whenever a message is published, but avoids the complication of modeling a subset of compromised recipients.

The meta fields of the tokens contains the following subfields: (1) stage, (2) position, (3) sigma. stage can have three possible values identifying three possible cases: (1) public key distribution, (2) share of the public-private key pair, (3) message. Using stage subfield, any party in the protocol recognizes if the message is part of keydist messages, or part of post messages. When the value of stage is message, the user posts $V_i(w)$, and position takes the value of $w$. sigma includes the identity of the sender and a proof of computation whenever necessary. sigma fields helps in the verify stage, we avoid the details here.

If we want to analyze the user distribution for $\Pi_{DC}$, we do not count the first round since it is used only for the key exchange and no user message is sent. Note that, if we get rid of the cryptographic primitives, we do not require the first round. If we assume that all the users are ready with their messages at the beginning, the latency overhead of $\Pi_{DC}$ is 1, and the bandwidth overhead is $\geq (N-1)$ per user.

**Modeling Tor.** This section demonstrates that onion routing protocols like Tor can be easily modeled using our Petri net model $M$. We want to stress that we focus on sender anonymity game against a global passive adversary, and hence, we do shall not model any sophisticated features like hidden services, congestion control etc.

Since Tor does not operate in rounds, embedding it into our model is not straightforward. Suppose, a Tor node takes at least $x$ milliseconds to process a message when it receives a message, and it takes at least $y$ milliseconds for a message to travel from one node to the next node over a network link. Then we define one round as $x + y$ milliseconds. Assume a perfect condition where each node takes exactly equal computation time for one message, and each link has exactly same delay. In the real world, delays and computation times are less stable, but can be estimated by an adversary. Instead of analyzing this, we instead allow the messages to remain within the node for the respective time.

Tor nodes and recipients are separate entities and hence, $S$, $P$ and $R$ are mutually exclusive. Whenever a Tor node receives a message, the node immediately processes and forwards that message to the next node or recipient. We can either model the latency overhead $\ell$ of Tor by estimating the time messages spend within the network that exceeds the (minimal) round length $x+y$ from above, or we set it to the number of hops, i.e., $\ell = 3$. In either case, we assume that $\ell$ does not increase with $\eta$ and thus get a latency overhead $\ell \in O(1)$. For analyzing Tor with a variable number $h$ of hops, we can instead set $\ell = h$. When a party $P_i$ receives a message, $T_{P_i}$ can retrieve the next hop from the meta field of the message. Since Tor does not add any noise messages, the bandwidth overhead is $\beta = 0$.

**C. Polynomial Boundedness of the adversary $A_{\text{paths}}$**

**Lemma 3.** If number of messages per user per round is polynomially bounded and the anonymity game stops in polynomial time, our adversary $A_{\text{paths}}$ is also polynomially bounded.
Proof Sketch. Suppose, the number of messages per user per round is bounded by $G$, and the game stops in $Q$ rounds. Then, total number of messages in the system is bounded by $(G \times N \times Q)$. Consequently, total number of elements in the set Tokens is bounded by $(G \times N \times Q \times Q)$. In the worst scenario, $A_{paths}$ will have to scan $(G \times N \times Q \times Q)$ tokens.

As long as the user distribution ensures that the number of messages per user per round is polynomially bounded, our adversary is polynomially bounded - which is true for both the user distributions that we use. For the synchronized user distribution, number of messages per user per round is $\frac{N+1}{N}$; and for the unsynchronized user distribution that is $p \leq 1$.

APPENDIX B

UNSYNCHRONIZED USERS WITH PARTIALLY COMPROMISING ADVERSARIES::LOWER BOUND ON ADVERSARIAL ADVANTAGE (CONT.)

Proof of Theorem 8. As in the proofs for Theorems 1, 3 and 6 we calculate the advantage of $A_{paths}$ against $\Pi_{ideal}$ to derive a bound against any protocol in our model.

As in the proof for Theorem 6 we define the random variables $X^{(1)}(x), X^{(2)}(x), \ldots, X^{(N)}(x)$, where $X^{(i)}(x)$ denotes the event of the $i^{th}$ user sending her own message in an interval of $x$ rounds $[a,b]$, with $(b-a) = x$. All $X^{(i)}(x)$ are mutually independent. Note that we here consider intervals that are not necessarily of size $\ell$.

$$X^{(i)}(x) = \begin{cases} 0 & \text{with probability} \quad (1-p)^x \\ 1 & \text{with probability} \quad (1-p)^x \end{cases}$$

As before, we make use of the sum $X(x) = \sum_{i=1}^{N} X^{(i)}(x)$ over all users and calculate the expected value of $X(x)$ as

$$E[X(x)] = E\left[\sum_{i=1}^{N} X^{(i)}(x)\right] = \sum_{i=1}^{N} E\left[X^{(i)}(x)\right] = N(1-(1-p)^x) = \mu(x)$$

Using the Chernoff Bound on the random variable $X(x)$ calculate $Pr \left[ X(x) - \mu(x) \geq Na \right] \leq \exp(-2a^2N)$, and for $a = \frac{\mu(x)}{N}$, we define $E(x)$ as:

$$E(x) = Pr \left[ X(x) \geq 2\mu(x) \right] \leq \exp\left( -2\left(1 - (1-p)^x\right)^2N \right)$$

Note that, similar to $X^{(i)}(x)$ and $X(x)$, $\mu(x)$ is also defined as in the proof for Theorem 6, but for a slice of variable length $x$. We denote the event that sender $u_{1-b}$ sends at least one message in an interval of size $x$ by $Y(x)$ and since all users are acting independently from each other we get for $j \in \{0, \ldots, N\}$, $Pr[Y(x) \mid X(x) = j] = 1 - Pr[\neg Y \mid X(x) = j] = \frac{1}{N}$. Moreover, for any value of $x$ with $2\mu(x) \leq N$,

$$Pr[Y(x)] = Pr[X(x) \geq 2\mu(x)] \times Pr[Y(x) \mid X(x) \geq 2\mu(x)]$$

$$+ Pr[X(x) < 2\mu(x)] \times Pr[Y(x) \mid X(x) < 2\mu(x)]$$

$$\leq Pr[X(x) \geq 2\mu(x)] \times Pr[Y(x) \mid X(x) = N]$$

$$+ Pr[X(x) < 2\mu(x)] \times Pr[Y(x) \mid X(x) = 2\mu(x)]$$

$$= E(x) \times Pr[Y(x) \mid X(x) = N]$$

$$+ (1 - E(x)) \times Pr[Y(x) \mid X(x) = 2\mu(x)]$$

$$= E(x) \left( \frac{n}{h} \right) + (1 - E(x)) \left( 2^{\mu(x)/h} \right)$$

$$= 1 - (1 - E(x)) \left( 1 - 2 \left( 1 - (1-p)^x \right) \right).$$

If $2\mu(x) > N$, we get with $f(x) = \min \left( \frac{1}{2}, 1 - (1-p)^x \right)$:

$$Pr[Y(x)] \leq E(x) + (1 - E(x)) \times 1 \leq 1$$

$$\leq 1 - (1 - E(x))(1 - 2f(x)).$$

Now, we calculate the probability of Invariant 1 being true, under our protocol $\Pi_{ideal}$ and as in the proof for Theorem 3. We distinguish two cases depending on $c$ and $\ell$:

Case 1): $c > \ell$

$$Pr[\text{Invariant 1 is true}]$$

$$\leq Pr[\neg \text{Cmpr}(\ell)] \times Pr[u_{1-b}.sent(r - \ell, r - c)]$$

$$+ Pr[\text{Cmpr}(\ell)] \times Pr[Y(\ell)]$$

$$\leq \left[ 1 - \left( \frac{1}{c} \right)^{\ell/c} \right] \left[ 1 - (1 - E(\ell)) \left( 1 - 2f_p(\ell) \right) \right].$$

By applying Markov’s inequality on the random variable $X(x)$, we get $E(x) = Pr[X(x) \geq 2\mu(x)] \leq \frac{1}{2}$. Thus, we derive for $\delta$:

$$\delta \geq 1 - \left[ 1 - \left( \frac{1}{c} \right)^{\ell/c} \right] \left[ \frac{1}{2} + f_p(\ell) \right].$$

Case 2a): $c < \ell$. As for the proof of Theorem 3 we split this case into two sub-cases, depending on $t$ and $c$.

Case 2a): $c < t$

$$Pr[\text{Invariant 1 is true}]$$

$$\leq Pr[u_{1-b}.sent(r - \ell, r - c)] + Pr[\neg u_{1-b}.sent(r - \ell, r - c)]$$

$$\times Pr[u_{1-b}.sent(r - c, r)] \times Pr[\neg \text{Cmpr}(c)]$$

$$= Pr[Y(\ell - c)] + Pr[Y(\ell - c)] \times Pr[\neg \text{Cmpr}(c)]$$

$$\leq \left[ 1 - (1 - E(\ell - c)) \left( 1 - 2f_p(\ell - c) \right) \right]$$

$$\times \left[ 1 - (1 - E(\ell - c)) \left( 1 - 2f_p(\ell - c) \right) \right] \left[ 1 - \left( \frac{1}{c} \right)^{\ell/c} \right].$$

Thus, for the adversarial advantage $\delta$ we derive,

$$\delta \geq 1 - Pr[\text{Invariant 1 is true}]$$

$$\geq 1 - \left[ (1 - E(\ell - c)) \left( 1 - 2f_p(\ell - c) \right) \right]$$

$$\times \left[ 1 - (1 - E(\ell - c)) \left( 1 - 2f_p(\ell - c) \right) \right] \left[ 1 - \left( \frac{1}{c} \right)^{\ell/c} \right].$$

We again use Markov’s inequality to replace $E(x)$ by $1/2$.

Case 2b): $t \leq c$

$$Pr[\text{Invariant 1 is true}]$$

$$\leq Pr[u_{1-b}.sent(r - \ell, r - c)] \times Pr[\neg \text{Cmpr}(t)]$$

$$+ Pr[\neg u_{1-b}.sent(r - \ell, r - c)]$$

$$\times Pr[u_{1-b}.sent(r - c, r)] \times Pr[\neg \text{Cmpr}(c)]$$

$$\leq Pr[u_{1-b}.sent(r - \ell, r - c)] + Pr[\neg u_{1-b}.sent(r - \ell, r - c)]$$

$$\times Pr[u_{1-b}.sent(r - c, r)] \times Pr[\neg \text{Cmpr}(c)]$$
The above event expression is exactly same as the expression we had in the previous case \((t > c)\). Thus, the rest of the calculations and bounds are exactly same as the previous case.

\[\text{APPENDIX C} \]

\textbf{RECIPIENT ANONYMITY}

The protocol model remains unchanged for recipient anonymity with the exception that the colored token now additionally has a private field for the recipient. We also require noise messages to adhere to the latency bound \(\ell\). Now we assume that there are \(N\) recipients in \(R\). Since, we are not concerned about distinguishing senders, we can assume that there is only one sender in \(\mathcal{S}\).

The anonymity game still remains almost same as Section IV-B, with only one change: now the game uses \(\alpha_{RA}\) instead of \(\alpha_{SA}\) as the anonymity notion. Naturally, the adversary is not informed about the delivery of the challenge message by a recipient, but of the sending of the challenge message by a sender.

The adversarial strategy \(A_{\text{paths}}\) also remains similar to that of sender anonymity scenario. But here, the adversary can identify the challenge message when it is sent by the sender, not when received by a recipient. After observing the tokens, \(A_{\text{paths}}\) tries to construct possible paths for the challenge message to the challenge recipients.

More formally, Given two challenge recipients \(R_0\) and \(R_1\) and the set of observed tokens \((t, r)\) in Tokens, where \(t\) is the token and \(r\) the round in which the token was observed, an adversary can construct the sets \(S_j\) (for \(j \in \{0, 1\}\)). Assume the challenge message is sent by the sender \(u\) in a round \(s\). We construct possible paths of varying length \(k\), s.t., each element \(p \in S_j\) represents a possible path of the challenge message starting from the sender \(u\) in round \(r_1 = s\) and the challenge message then arrives at \(R_j\) (\(j \in \{0, 1\}\)) in round \(r_k\). With challenge bit \(b\), \(S_b\) cannot be empty, as the actual path taken by the challenge message has to be one element in \(S_b\).

\[S_j = \{p = (t_1, \text{prev} \dots, t_k, \text{prev}, t_k, \text{next}) : ((t_1, r_1), \ldots, (t_k, r_k)) \in \text{Tokens} \text{ s.t.}
\]
\[
t_1, \text{prev} = u \land \text{t}_k, \text{next} = R_j
\]
\[
\land t_1, \text{msg} = \text{Challenge} \land k \leq \ell
\]
\[
\land \forall i \in \{1, \ldots, k-1\} (t_i, \text{next} = t_{i+1}, \text{prev} \land r_{i+1} = r_i + 1
\]
\[
\land (\exists t_{i+1} : (t_{i+1}, r_{i+1}) \in \text{Tokens} \land t_{i+1}, \text{prev} = t_i, \text{next} \land t_{i+1}, \text{ID}_t = t_i, \text{ID}_t = t_{i+1} = t_{i+1}))
\]

**Necessary invariant for recipient anonymity.** For recipient anonymity it is necessary that at least both challenge recipients receive messages in the \(\ell\) rounds after the challenge message was sent. Moreover, on the path of the actual challenge message, there needs to be at least one honest (non-compromised) party, as otherwise the adversary can track the challenge message from the sender to the recipient (\(S_b\) will have exactly one element and \(S_{1-b}\) will be empty). Those two conditions together form our necessary protocol invariant.

\[\textbf{Invariant 2.} \text{ Let } R_0 \text{ and } R_1 \text{ be the challenge recipients; let } b \text{ be the challenge bit; and let } s \text{ be the time when the sender } u \text{ sends the challenge message towards } R_0. \text{ Assume that messages for } R_{1-b} \text{ (including noise messages) are received by } R_{1-b} \text{ at times } V_{RA} = \{t_1, t_2, t_3, \ldots, t_k\}. \text{ Now, let } T_{RA} = \{t : t \in V_{RA} \land s < t \leq (s + \ell)\}. \text{ Then,}
\]
\[(i) \text{ the set } T_{RA} \text{ is not empty, and}
\]
\[(ii) \text{ the challenge message passes through at least one honest node at some time } t' \text{ such that } s \leq t' \leq \max(T_{RA}).\]

The invariant is very similar to Invariant 1 with the only difference that we consider messages sent towards recipients (instead of messages sent by users). In contrast, for sender anonymity, where sending messages was the main criteria, for recipient anonymity analogously receiving messages is the main criteria and the times at which messages are received can be (partially) controlled by the protocol.

**Claim 4** (Invariant 2 is necessary for anonymity). Let \(\Pi\) be any protocol in \(M\) with latency overhead \(\ell\) and bandwidth overhead \(\beta\). Let \(u, R_0, R_1, b\) and \(T_{RA}\) be defined as in Invariant 2. If Invariant 2 is not satisfied, then our adversary \(A_{\text{paths}}\) as in Definition 5 wins (against recipient anonymity).

**Proof.** We distinguish two cases, depending on \(T_{RA}\): either \(T_{RA}\) is empty, or \(T_{RA}\) is non-empty.

If the set \(T_{RA}\) is empty, then \(S_{1-b}\) is empty as well. However, by construction of our protocol model, the set \(S_b\) is always non-empty. Consequently, the adversary \(A_{\text{paths}}\) will output \(b\) and thus win with probability 1. If \(T_{RA}\) is not empty, the following cases can occur:

1) The challenge message never passes through an honest node: In this case, the field \(ID_t\) of the message never changes for the tokens. Thus, \(S_b\) will have exactly one element, and \(S_{1-b}\) will be an empty set, and consequently \(A_{\text{paths}}\) wins.

2) The challenge message passes through one or more honest nodes at times \(t'\), such that \(t' > \max(T_{RA})\), but not before: Following the same reasoning as above, we see that paths before \(\max(T_{RA})\) can be ambiguous, but none of them leads to \(R_{1-b}\). Hence, \(S_b\) can have multiple elements, but \(S_{1-b}\) will still be an empty set. Thus, \(A_{\text{paths}}\) wins.

3) The challenge message passes through an honest node at time \(t'\) with \(t' \leq \max(T_{RA})\): In this case, the invariant is true.

In all of the above mentioned cases either the invariant is true, or the adversary wins with probability 1.

**Claim 5** (Internally terminated noise does not influence Invariant 2). Any message that is not delivered to a recipient \(R \in R\) does not influence the probability for Invariant 2 being true.

The proof for this claim is analogous to the proof for Claim 2, where instead of considering the sending of messages, we are concerned with receiving messages.

**Ideal Protocol.** Now we construct our ideal protocol \(\Pi_{\text{ideal}}\), which is very similar to that of the sender anonymity scenario.
The routing strategy is exactly the same as the sender anonymity scenario. Even for deciding the optimal delivery time $t$ for each message, the protocol queries oracle $O$. The oracle $O$ returns the values in such a way that it maximizes the probability of satisfying Invariant 2 for the given routing strategy. The only difference here is that, in sender anonymity scenario the optimal delivery time $t$ for each message is independent of the optimal delivery times for other messages, but here they are dependent.

**Claim 6** (Ideal protocol is ideal for the invariant). $\Pi_{\text{ideal}}$ satisfies Invariant 2 with a probability at least as high as any other protocol in $M$, against the given adversary $A_{\text{paths}}$.

**Proof.** We want to prove our claim by contradiction. For contradiction, we assume that $\Pi_{\text{ideal}}$ is not the best protocol. That means, there exist a protocol $\Pi_{\text{new}}$ which satisfy Invariant 2 with a higher probability than $\Pi_{\text{ideal}}$.

We construct a protocol $\Pi_{\text{hybrid}}$ which exactly follows the routing strategy of $\Pi_{\text{ideal}}$, but for each message $\Pi_{\text{hybrid}}$ selects the time delay same as $\Pi_{\text{new}}$ instead of querying from oracle $O$.

Let’s compare the protocols $\Pi_{\text{hybrid}}$ and $\Pi_{\text{new}}$. Since both the protocol select the same time delays for every message, the set $T_{RA}$ is exactly same for both of them. Consequently, the path length for the challenge message till $\max(T_{RA})$ is same for both the protocols. But, $\Pi_{\text{hybrid}}$ maximizes the number of distinct parties on the path, hence maximizes the probability of having at least one honest party on the path. Therefore, $\Pi_{\text{hybrid}}$ has a probability of satisfying Invariant 2 at least as high as $\Pi_{\text{new}}$.

Next, let’s compare $\Pi_{\text{hybrid}}$ and $\Pi_{\text{ideal}}$: They both employ the same routing strategy. But $\Pi_{\text{ideal}}$ selects the time delays for all messages by querying oracle $O$. By definition of the oracle, it selects the time delays in such a way that it maximizes the probability of satisfying Invariant 2, for the given routing strategy. Hence, $\Pi_{\text{ideal}}$ satisfies Invariant 2 with a probability at least as high as $\Pi_{\text{hybrid}}$. Thus, $\Pi_{\text{new}}$ does not satisfy Invariant 2 with a probability higher than $\Pi_{\text{ideal}}$ - which contradicts our initial assumption that $\Pi_{\text{ideal}}$ is not the best protocol. \qed

### A. Recipient Anonymity of Synchronized Users with Non-compromising Adversaries

As for our first scenario for sender anonymity, we investigate an ideal user distribution where inputs from all users are globally synchronized.

We assume that all the input messages come within $N'$ rounds, exactly one message per round, following a random permutation that assigns one round to each recipient. Formally we group together all users that send messages into one sender that sends all the messages. In a given round, the sender should send a message for the assigned recipient. Then, the protocol decides when to deliver the message to the recipient, but not delaying more than $\ell$ rounds.

We denote this user distribution with $U_B$. Since, we are considering a globally controlled user distribution, we are considering a globally controlled noise as well. The protocol can add a maximum of $B = \beta N'$ noise messages per round, or $\beta$ noise messages per recipient per round, where $0 \leq \beta \leq 1$. We consider a non-compromising passive adversary that can observe all network traffic.

**Theorem 9.** For user distribution $U_B$, no protocol $\Pi \in M$ can provide $\delta$-recipient anonymity for any $\delta < 1 - f^R_{A}(\ell)$, where $f^R_A(d) = \min\left(1, \frac{(\ell + d) + (\ell + d)\beta N'}{N'}\right)$.

**Proof.** By Claim 6, we know that $\Pi_{\text{ideal}}$ has the highest probability to satisfy Invariant 2 against $A_{\text{paths}}$. Thus, by Claim 4 it suffices to calculate the probability for $\Pi_{\text{ideal}}$ to satisfy the invariant as a lower bound of the adversary’s advantage against any protocol.

Let, $R_0$ and $R_1$ be the recipients chosen by the adversary and let $b$ be the challenge bit. Let $s$ be the round in which the sender sends the challenge message.

Recall that Invariant 2 is necessary for the protocol to provide anonymity. Since we are considering a non-compromising adversary, $Pr[\text{Invariant 2 is true}] = Pr[T \text{ is not empty}]$. If a message is sent for the recipient $R_{1-b}$ (enters the protocol) in $[s - \ell, s + \ell - 1]$, it has a possibility to populate an element in $T_{RA}$. With the above in mind, let us define the following events:

- $H_1$: Within $2\ell$ rounds a noise message is sent to $R_{1-b}$.
- $H_2$: Within $2\ell$ rounds a user sends a real message to $R_{1-b}$.
- $H_3$: Invariant 2 is true.

We proceed analogously to the proof for Theorem 1 and get:

$$Pr[H_2] \leq \frac{2\ell}{N'}.$$

Similarly, in each round noise messages are sent to $\beta N$ unique users in such a way that no real message is scheduled for them. Thus, $Pr[H_1] \leq \frac{2\ell\beta N'}{N}$.

We combine these insights to yield a bound.

$$Pr[H_3] = Pr[H_1 \lor H_2] = \min(1, Pr[H_1] + Pr[H_2]) \leq \min(1, \frac{2\ell + 2\ell\beta N'}{N}).$$

Thus, we can bound the probability for the adversary as $Pr[0 = A_{\text{paths}}, b = 1] = Pr[1 = A_{\text{paths}}, b = 0] \leq \frac{1}{2} Pr[H_3]$; and $Pr[0 = A_{\text{paths}}, b = 0] \geq 1 - \frac{1}{2} Pr[H_3]$. And therefore, since $\delta \geq Pr[0 = A_{\text{paths}}, b = 0] - Pr[0 = A_{\text{paths}}, b = 1], \delta \geq 1 - Pr[H_3] \geq 1 - f^R_A(\ell)$. \qed

**Impossibility for Strong Recipient Anonymity.** We now investigate under which constraints for $\ell$ and $\beta$ Theorem 9 rules out strong recipient anonymity.

**Theorem 10.** For user distribution $U_B$ with $\ell < N'$ and $\beta N' \geq 1$, no protocol in $M$ can achieve strong recipient anonymity if $4\ell\beta < 1 - \epsilon(\eta)$, where $\epsilon(\eta) = \frac{1}{\eta^d}$ for a positive constant $d$.

The proof follows analogously to the proof of Theorem 2.
B. Recipient Anonymity of Synchronized Users with Partially Compromising Adversaries

Now we extend our analysis for recipient anonymity against partially compromising adversaries, with the same user distribution as the previous section.

**Theorem 11.** No protocol $\Pi \in M$ can provide $\delta$-recipient anonymity for the user distribution $U_B$, where

$$\delta < \begin{cases} 1 - \left(1 - \frac{\ell_c}{\ell}\right) f_{\beta}^{RA}(\ell) & c \geq \ell \\
1 - \left(1 - \frac{\ell}{\ell_c}\right) f_{\beta}^{RA}(c) - f_{\beta}^{RA}(\ell - c) & c < \ell \\
\end{cases}$$

where $f_{\beta}^{RA}(d) = \min \left(1, \left(\frac{(f + d) + (f + d)\beta N'}{N'}\right)\right)$.

**Proof.** Let $R_0, R_1$ be the challenge users and let $b$ be the challenge bit. Moreover, let $s_0$ be the time the challenge message is sent for $R_0$ and let $r = s_0 + t$ be the time it is received by the recipient, where $t$ is the delivery time decided by the oracle $O$ for the challenge message.

We distinguish two cases, depending on $\ell$ and $c$: 1) First, where the number of compromised parties $c$ is at least as large as the maximal latency $\ell$. In this case, all parties on the path of the challenge message could be compromised. 2) Second, where all parties on the path of the challenge message cannot be compromised. And hence, the analysis focuses on the delivery times of messages for $R_{1-b}$.

1) **Case $c \geq \ell$.** We know, $t \geq t$ holds by definition. The invariant is true if and only if $R_{1-b}$ receives at least one message in one of the rounds between $(s_0 + 1)$ and $(s_0 + \ell)$ and for the last of those messages, delivered at time $t_{lst}$, there is at least one non-compromised party on the path between $t_0$ and $t_{lst}$. Hence,

$$\Pr_{[\text{Invariant 2 is true}] = \Pr[R_{1-b} \text{ receives at least one message in } [s_0, s_0 + \ell]] \times \Pr[NOT \text{ all the } c \text{ parties are compromised}] \leq f_{\beta}^{RA}(\ell) \left(1 - \frac{\ell_c}{\ell}\right).$$

Hence, $\delta \geq 1 - \left(1 - \frac{\ell_c}{\ell}\right) f_{\beta}^{RA}(\ell)$

2) **Case $c \leq \ell$:***

The probability that all parties on the mutual path of the challenge message and a message for the alternative recipient $R_{1-b}$ are compromised now mainly depends on the delivery time of the messages for $R_{1-b}$. We distinguish two sub-cases depending on the oracle’s choice for $t$:

2a) **Case $c \leq t$:***

$$\Pr_{[\text{Invariant 2 is true}] \leq \Pr[R_{1-b} \text{ receives at least one message in } [s_0 + c, s_0 + \ell]] \times \Pr[R_{1-b} \text{ does NOT receive a message in } [s_0 + c, s_0 + \ell]] \times \Pr[R_{1-b} \text{ receives at least one message in } [s_0, s_0 + c]] \times \Pr[NOT \text{ all the } c \text{ parties are compromised}] \leq f_{\beta}^{RA}(\ell - c) + f_{\beta}^{RA}(c) \left(1 - \frac{1}{\ell_c}\right).$$

Hence, $\delta \geq 1 - \left(1 - \frac{1}{\ell_c}\right) f_{\beta}^{RA}(\ell - c) - f_{\beta}^{RA}(\ell - c)$.

2b) **Case $t \leq c$:***

$$\Pr_{[\text{Invariant 2 is true}] \leq \Pr[R_{1-b} \text{ receives at least one message in } [s_0 + c, s_0 + \ell]] \times \Pr[NOT \text{ all the } t \text{ parties are compromised}] + \Pr[R_{1-b} \text{ does NOT receive any message in } [s_0, s_0 + \ell]] \times \Pr[R_{1-b} \text{ receives at least one message in } [s_0, s_0 + c]] \times \Pr[NOT \text{ all the } t \text{ parties are compromised}] \leq \Pr[R_{1-b} \text{ receives at least one message in } [s_0 + c, s_0 + \ell]] + \Pr[R_{1-b} \text{ does NOT receive any message in } [s_0, s_0 + \ell]] \times \Pr[R_{1-b} \text{ receives at least one message in } [s_0, s_0 + c]] \times \Pr[NOT \text{ all the } t \text{ parties are compromised}]$$

The above event expression is exactly the same as the expression we had in the previous case ($t > c$). The bound on $\delta$ thus follows analogously.

**Impossibility for Strong Recipient Anonymity.** We now investigate under which constraints for $c$, $\ell$ and $\beta$ Theorem 9 rules out strong recipient anonymity.

**Theorem 12.** For user distribution $U_B$ with $K \in poly(\eta)$, $K > c \geq \ell$, $\ell < N'$ AND $\beta N' \geq 1$, no protocol can achieve strong anonymity if $4\beta < 1 - \epsilon(\eta)$ OR $\ell \in O(1)$, where $\epsilon(\eta) = 1/\eta^d$ for a positive constant $d$.

The proof follows analogously to the proof of Theorem 4.

**Theorem 13.** For user distribution $U_B$ with $K \in poly(\eta)$, constant $c$, $K > \ell > c$, $\ell < N$ AND $\beta N \geq 1$, no protocol can achieve strong anonymity if $4(\ell - c)\beta < 1 - \epsilon(\eta)$, where $\epsilon(\eta) = 1/\eta^d$ for a positive constant $d$.

The proof follows analogously to the proof of Theorem 5.

C. Recipient Anonymity of Unsynchronized Users with Non-compromising Adversaries

Now we shall consider unsynchronized user distribution, which is similar to the unsynchronized user distribution for sender anonymity, but with a few changes. Our unified sender has a biased coin corresponding to each recipient with success probability $p$. In each round, he decides to send a message to a recipient by tossing the biased coin, independent of other recipients as well as other rounds. We denote this user distribution with $U_P$. We consider a non-compromising passive adversary similar to Section C-A.

**Theorem 14.** For user distribution $U_P$, no protocol $\Pi \in M$ can provide $\delta$-recipient anonymity for any $\delta < 1 - \left(1 - \frac{1}{p + f_{P}^{RA}(\ell)}\right)$, where $f_{P}^{RA}(d) = \min \left(\frac{1}{p}, 1 - (1 - p)^{\ell + d}\right)$ for integer $d \geq 1$.

**Proof.** By Claim 6, we know that $\Pi_{\textit{deal}}$ is the optimal protocol for satisfying Invariant 2 and by Claim 4 we know that the invariant is necessary for anonymity. Thus, it suffices to
calculate the probability that $\Pi_{\text{ideal}}$ satisfies Invariant 2 as a lower bound of the adversary’s advantage against any protocol.

Let, $R_0$ and $R_1$ be the recipients chosen by the adversary and let $b$ be the challenge bit. Let $s$ be the round in which the sender sends the challenge message.

Since we are considering a non-compromising adversary, $\Pr[\text{Invariant 2 is true}] = \Pr[\mathcal{T}_{RA} \text{ is not empty}].$ Note that, if a message is sent for the recipient $R_{1-b}$ (enters the protocol) in $[s-\ell, s+\ell-1]$, it has a possibility to populate an element in $\mathcal{T}_{RA}$.

We follow the same calculations as in the proof of Theorem 6, and derive:

$$\Pr[\mathcal{T}_{RA} \text{ is not empty}] \leq \Pr[Y(d)] = 1 - (1 - E(d)) (1 - 2f_p(d)),$$

Where $f_p(d)$ is defined as in Theorem 6, and $Y(d)$ denotes the event that at least one message is sent for a given recipient within an interval of $d$ rounds.

We distinguish two cases, depending on $c$: 1) First, where the number of compromised parties $c$ is at least as large as the maximal latency $\ell$. In this case, all parties on the path of the challenge message could be compromised. 2) Second, where all parties on the path of the challenge message cannot be compromised. And hence, the analysis focuses on the delivery times of messages for $R_{1-b}$.

1) Case $c \geq \ell$. We know, $\ell \geq t$ holds by definition. The invariant is true if and only if $R_{1-b}$ receives at least one message in one of the rounds between $(s_0 + 1)$ and $(s_0 + \ell)$ and for the last of those messages, delivered at time $t_{last}$, there is at least one non-compromised party on the path between $t_0$ and $t_{last}$. Hence,

$$\Pr[\text{Invariant 2 is true}] \leq \Pr[R_{1-b} \text{ receives at least one message in } [s_0, s_0 + \ell]] \times \Pr[NOT \text{ all the c parties are compromised}]$$

$$\leq \Pr[Y(2\ell)] \times \left[1 - \frac{f_p(2\ell)}{2}\right] = \left[1 - \frac{1}{2} + f_p(2\ell)\right] \left[1 - \frac{\ell}{2}\right].$$

Therefore, $\delta \geq 1 - \Pr[\mathcal{T}_{RA} \text{ is not empty}] \geq 1 - \left[\frac{1}{2} + f_p(2\ell)\right]$.

**Impossibility for Strong Recipient Anonymity.** We now investigate under which constraints for $\ell$ and $\beta$ Theorem 9 rules out strong recipient anonymity.

Theorem 15. For user distribution $U_p$ and $p > 0$, no protocol can achieve strong anonymity recipient if $2\ell p < 1 - \epsilon(\eta)$, where $\epsilon(\eta) = 1/\eta^d$ for a positive constant $d$.

The proof follows analogously to the proof of Theorem 7.

D. Recipient Anonymity of Unsynchronized Users with Partially Compromising Adversaries

Now we extend our analysis for recipient anonymity against partially compromising adversaries, with the same user distribution as the previous section.

Theorem 16. No protocol $\Pi \in M$ can provide $\delta$-recipient anonymity for the user distribution $U_p$, when

$$\delta < \left\{\begin{array}{ll}
1 - \left[\frac{f_p(\ell)}{2}\right] & \text{for } c \geq \ell \\
\left[\frac{1}{2} + f_p(\ell - c)\right] \left(1 - \left[\frac{1}{2} + f_p(\ell - c)\right]\left[1 - \frac{1}{2}\right]\right) & \text{for } c < \ell
\end{array}\right.$$\

where $f_p(d) = \min \left(\frac{1}{2}, 1 - (1-p)^{\ell+d}\right)$ for integer $d \geq 1$.

**Proof.** Let $R_0, R_1$ be the challenge users and let $b$ be the challenge bit. Moreover, let $s_0$ be the time the challenge message is sent for $R_0$, and let $t = s_0 + t$ be the time it is received by the recipient, where $t$ is the delivery time decided by the oracle $O$ for the challenge message.

As in proofs for Theorems 8 and 14, we define $Y(d)$ as the event that at least one message is sent for a given recipient within an interval of $d$ rounds; and we derive:

$$\Pr[Y(d)] \leq 1 - \frac{1}{2} (1 - 2f_p(d)) = \frac{1}{2} + f_p(\ell)\left[\frac{1}{2}\right].$$

We distinguish two cases, depending on $\ell$ and $c$: 1) First, where the number of compromised parties $c$ is at least as large as the maximal latency $\ell$. In this case, all parties on the path of the challenge message could be compromised. 2) Second, where all parties on the path of the challenge message cannot be compromised. And hence, the analysis focuses on the delivery times of messages for $R_{1-b}$.

1) Case $c \geq \ell$. We know, $\ell \geq t$ holds by definition. The invariant is true if and only if $R_{1-b}$ receives at least one message in one of the rounds between $(s_0 + 1)$ and $(s_0 + \ell)$ and for the last of those messages, delivered at time $t_{last}$, there is at least one non-compromised party on the path between $t_0$ and $t_{last}$. Hence,

$$\Pr[\text{Invariant 2 is true}]$$

$$\leq \Pr[R_{1-b} \text{ receives at least one message in } [s_0, s_0 + \ell]] \times \Pr[NOT \text{ all the c parties are compromised}]$$

$$\leq \Pr[Y(2\ell)] \times \left[1 - \frac{f_p(2\ell)}{2}\right] = \left[1 - \frac{1}{2} + f_p(2\ell)\right] \left[1 - \frac{\ell}{2}\right].$$

Therefore, $\delta \geq 1 - \Pr[\mathcal{T}_{RA} \text{ is not empty}] \geq 1 - \left[\frac{1}{2} + f_p(2\ell)\right]$.

2) Case $c < \ell$. The probability that all parties on the mutual path of the challenge message and a message for the alternative recipient $R_{1-b}$ are compromised now mainly depends on the delivery time of the messages for $R_{1-b}$.

**2a) Case $c \leq t$:**

$$\Pr[\text{Invariant 2 is true}]$$

$$\leq \Pr[R_{1-b} \text{ receives at least one message in } [s_0 + c, s_0 + \ell]] + \Pr[R_{1-b} \text{ does NOT receive a message in } [s_0 + c, s_0 + \ell]]$$

$$\times \Pr[R_{1-b} \text{ receives at least one message in } [s_0, s_0 + c]]$$

$$\times \Pr[NOT \text{ all the c parties are compromised}]$$

$$\leq \Pr[Y(2\ell - c)] + (1 - \Pr[Y(2\ell - c)])$$

$$\times \Pr[Y(\ell + c)] \left[1 - \frac{1}{2}\right].$$

Therefore, since $\delta \geq 1 - \Pr[\text{Invariant 2 is true}]$, we have:

$$\delta \geq \left(1 - \Pr[Y(2\ell - c)]\right) \left(1 - \Pr[Y(\ell + c)] \left[1 - \frac{1}{2}\right]\right)$$

$$\geq \left(1 - \left[\frac{1}{2} + f_p(2\ell - c)\right]\right) \left[1 - \left[\frac{1}{2} + f_p(\ell + c)\right]\left[1 - \frac{1}{2}\right]\right]$$

$$\geq \left(1 - \left[\frac{1}{2} + f_p(\ell - c)\right]\right) \left[1 - \left[\frac{1}{2} + f_p(\ell - c)\right]\left[1 - \frac{1}{2}\right]\right].$$
2b) Case \( t < c \) :

\[
\Pr \left[ \text{Invariant 2 is true} \right] \\
\leq \Pr \left[ R_{1-b} \text{ receives at least one message in } [s_0 + c, s_0 + \ell] \right] \\
\quad \times \Pr \left[ \not \text{ all the } t \text{ parties are compromised} \right] \\
+ \Pr \left[ R_{1-b} \text{ does NOT receive any message in } [s_0, s_0 + \ell] \right] \\
\quad \times \Pr \left[ R_{1-b} \text{ receives at least one message in } [s_0, s_0 + c] \right] \\
\quad \times \Pr \left[ \not \text{ all the } t \text{ parties are compromised} \right]
\]

\[
\leq \Pr \left[ R_{1-b} \text{ receives at least one message in } [s_0 + c, s_0 + \ell] \right] \\
+ \Pr \left[ R_{1-b} \text{ does NOT receive any message in } [s_0, s_0 + \ell] \right] \\
\quad \times \Pr \left[ R_{1-b} \text{ receives at least one message in } [s_0, s_0 + c] \right] \\
\quad \times \Pr \left[ \not \text{ all the } t \text{ parties are compromised} \right]
\]

The above event expression is exactly the same as the expression we had in the previous case \( t > c \). The bound on \( \delta \) thus follows analogously.

\[\square\]

E. Impossibility for Strong Anonymity

The bound for \( \delta \) is in this scenario is exactly similar to the counterpart of sender anonymity results (Section VIII). Hence, the analysis follows analogously.

APPENDIX D

VISUAL 3D REPRESENTATIONS OF THE RESULTS

In the paper, we focus on lower-bound results for strong anonymity (or negligible \( \delta \) values). However, our key Theorems 1, 3, 6 and 8 also offer lower bounds for non-negligible \( \delta \) values, which can be of interest to several AC protocols.

On our project webpage [33], we visualize these lower bounds using interactive 3D surface plots. In particular, we plot the adversarial advantage \( \delta \in [0, 1] \) as a function of \( \beta \) and \( \ell \). We encourage the readers to interact with these plots to better understand our results for non-negligible \( \delta \) values.

Here, in Figures 8 to 11, we present and analyze four snapshots of those lower bound plots for the number of users \( N = 10000 \). The x-axis represents latency \( \ell \) (ranging from 0 to 200), and the y-axis bandwidth overhead \( \beta \) (ranging from 0.0 to 0.04). But in Figure 10 and Figure 11, the y-axis actually represents total bandwidth \( p = p' + \beta \) as in Theorem 7. For curves in Figure 9 and Figure 11 we chose \( K = 100 \) as the number of total parties and \( c = 20 \) compromised parties.

A derived \( \delta \) lower bound for the non-compromising adversary is also a valid lower bound for a (partially) compromising adversary. For some edge cases (e.g., when \( \ell \) is close to \( N \) and \( \beta \) is close to 0), due to some approximations employed in the compromising adversaries scenario, the non-compromising adversary lower bound is actually tighter than the compromising adversaries lower bound. Therefore, in Figure 11, while plotting the 3D graph for a partially compromising adversary scenario, we have used the maximum of the lower bounds on \( \delta \) for compromising adversary and non-compromising adversary.

In each plot, the dark blue region indicates the possibility of obtaining strong anonymity. For any point \((x, y)\) outside those regions, strong anonymity is not possible. For example, as shown in Figure 8, for \( \ell = 100 \) the bandwidth overhead \( \beta \) has to be at least 0.01 to expect strong anonymity.

For the chosen \( c \) and \( K \), the plots in Figures 8 and 9 are almost identical as the \( \ell \) and \( \beta \) factors contribute more to anonymity than the compromised parties can affect it. If we instead compare Figure 10 with Figure 11, the effect of compromisation is noticeable: the dark blue region in Figure 11 is much smaller than that in Figure 10. Also, we can see a steep wall in Figure 11 for \( \ell \leq c = 20 \), demonstrating that providing anonymity becomes difficult when \( \ell < c \); however, for \( \ell > c \), the effect of compromisation is less noticeable.
Fig. 8. Synchronized User Distribution with Non-compromising Adversaries. $z = 1 - f_\beta(\ell)$, where $f_\beta(x) = \min(1, ((x - \beta N)/N - 1))$.

Fig. 9. Synchronized User Distribution with Partially compromising Adversaries. Total protocol parties $K = 100$, number of compromised parties $c = 20$. $z = 1 - [1 - (\ell/c)]/\beta/\ell$ for $\ell \leq c$, $z = 1 - (\ell - c)/\beta/\ell$ for $\ell > c$.

Fig. 10. Unsynchronized User Distribution with Non-compromising Adversaries. $z = 1 - (\ell + f_\beta(\ell))$, where $f_\beta(x) = \min(\ell/2, 1 - (1 - p)^x)$.

Fig. 11. Unsynchronized User Distribution with Partially compromising Adversaries. Total number of protocol parties $K = 100$, number of compromised parties $c = 20$. $z' = 1 - [1 - (\ell/c)]/[\ell + f_\beta(\ell)]$ for $\ell \leq c$, $z' = (1 - [1 - (\ell/c)]/[\ell + f_\beta(\ell)] \times (1 - [\ell - c]/f_\beta(\ell - c))$ otherwise. We set $z = \max(z', 1 - (\ell/2 + f_\beta(\ell)))$. 

26