Debajyoti Das*, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate

Comprehensive Anonymity Trilemma: User Coordination is not enough

Abstract: For anonymous communication networks (ACNs), Das et al. recently confirmed a long-suspected trilemma result that ACNs cannot achieve strong anonymity, low latency overhead and low bandwidth overhead at the same time. Our paper emanates from the careful observation that their analysis does not include a relevant class of ACNs with what we call *user coordination* where users proactively work together towards improving their anonymity. We show that such protocols can achieve better anonymity than predicted by the above trilemma result. As the main contribution, we present a stronger impossibility result that includes all ACNs we are aware of. Along with our formal analysis, we provide intuitive interpretations and lessons learned. Finally, we demonstrate qualitatively stricter requirements for the Anytrust assumption (all but one protocol party is compromised) prevalent across ACNs.

Keywords: anonymity, trilemma

DOI 10.2478/popets-2020-0056
Received 2019-11-30; revised 2020-03-15; accepted 2020-03-16.

1 Introduction

Anonymous communication networks (ACNs)[4, 7–10, 13, 16, 17, 22–24, 28, 31–33] are critical to communication privacy over the Internet as they enable individuals to maintain their privacy from untrusted intermediaries and endpoints. Typically, ACNs involve messages traveling through some intermediaries before arriving at their destinations, and therefore they introduce network latency and bandwidth overheads. Even after almost four decades of work, the search for an optimal overhead ACN design is still unfinished: Anonymity requires the company of others interested in anonymity, which indeed makes it a hard problem to solve. In practice, we continue to rely on low latency and bandwidth overhead networks such as Tor [29] that don't provide strong anonymity guarantees against global adversaries as the protocol designs offering strong anonymity [7, 10, 31, 33] (i.e., anonymity up to a negligible chance) incur a significant latency overhead, bandwidth overhead, or both. Although some of the recent designs [28] have been successful in reducing these overheads and also in offering latency-bandwidth trade-offs to an extent, it remains unclear how optimal these designs are. In our search for an optimal overhead ACN, lower bounds on the communication overhead play an important guiding role.

Recently, Das et al. [12] reduced the search space of possible ACN designs that could achieve strong anonymity by offering necessary constraints (i.e., impossibility results) for ACNs that relate bandwidth overhead, latency overhead, and the degree (or strength) of anonymity. They show a *universal necessary constraint* that states strong anonymity is impossible without significant bandwidth overhead, latency overhead, or an inversely proportional combination of those, even if no protocol parties are compromised. If some protocol parties are compromised, Das et al. showed an additional *necessary constraint* that in essence proves that strong anonymity excludes low latency (i.e., constant w.r.t. to the system's security parameter), regardless of the bandwidth overhead.

This necessary constraint arises from a key anonymity invariant (or restriction) that strong anonymity is possible only if messages from two honest users mix in an honest node. This invariant is in line with many mixnet designs from the literature [9, 13, 23, 28, 32]. However, protocols following the dining cryptographer networks (DC-nets) [7] design avoid this constraint by providing robustness against compromisation *using some pre-established coordination among users*. This leaves hope for ACNs that leverage such techniques.

1.1 Our contribution

In this work, we limit the hope left for ACNs. To this end we generalize techniques from the line of works [4, 7, 10, 16, 33] that escape the earlier impos-

***Corresponding Author: Debajyoti Das:** Purdue University, E-mail: das48@purdue.edu
Sebastian Meiser: Visa Research, E-mail: smeiser@visa.com
Esfandiar Mohammadi: Universitaet zu Luebeck, E-mail: esfandiar.mohammadi@uni-luebeck.de
Aniket Kate: Purdue University, E-mail: aniket@purdue.edu

sibility results of [12] in the following property, which we call user coordination: *assuming some form of (free) coordination among a set of N users, $h+1 \leq N$ users send a packet each for some single (actual) message such that (i) the receiver can retrieve the actual message only after receiving all the $h+1$ packets, and that (ii) the receiver of the message cannot distinguish who among the $h+1$ users actually sent the message.*

The goal here is not to keep user coordination practical; rather, we define the notion in this way to capture all efficient instantiations of similar techniques. One prominent example is given by DC-nets. DC-nets use shared keys/coins to produce dummy messages (corresponding to our shares) that allow the receiver to reconstruct the actual message.

We show that even protocols with user coordination must either use an excessive bandwidth overhead (every user sends a share for every real message by any other user) or adhere to our improved anonymity trilemma.

Formal lower bounds. Assuming that the bandwidth is not trivially high, we derive various lower bounds on the necessary latency overhead required for strong anonymity depending on the number c of compromised nodes. For $\mathsf{c} > 0$, we show that strong anonymity is impossible for constant latency ($\ell \in \Theta(1)$). If half of all K nodes are compromised ($\mathsf{c} = \mathsf{K}/2$), ACNs with strong anonymity cannot have a latency that is logarithmic ($\ell \leq \log(\mathsf{K})$) in the number K of protocol nodes. For the Anytrust setting [33] where all but a constant amount of protocol nodes are compromised, strong anonymity requires a minimal latency in the order of $\sqrt{\mathsf{K}}$.

Proof formalism. While basing our techniques on the foundation of the existing anonymity trilemma, we contribute several key elements to their technique that might be of independent interest: novel necessary constraints for anonymity as well as novel design goals for an ideal AC protocol. Moreover, we provide intuitive readings of our formal theorems that summarize their key insights.

Lessons learned. Our exercise in formally analyzing protocols with user coordination presents us with several tangible lessons: first, as our novel necessary constraints inherently are more lenient (they allow more anonymity with the same latency overhead and bandwidth overhead) our results thereby point future research on designing ACNs with reduced overhead in the direction of ACNs with user coordination, in particular with dynamic user coordination that relaxes the strict turn-by-turn scheduling of DC-nets and its several extensions. Second, we further contribute to the quest for optimal ACNs by clearly identifying the limits of our novel necessary constraints. Effectively, our necessary constraints raise open problems whose solutions would escape our results and could lead to ACNs that are very close to the universal necessary constraint.

Moreover, in Appendix A, we discuss the scope of our results by discussing and quantifying the complexity of user coordination techniques found in the literature.

1.2 Related work

Das et al. [12] formalized and confirmed the anonymity trilemma for mixing based protocols. They formally proved for which parameters of bandwidth overhead and latency overhead *strong anonymity* (anonymity up to a negligible adversarial advantage) is impossible. Their analysis, however, does not fully apply to a large class of protocols that includes DC-nets or secret sharing based protocols.

In prior work, Oya et al. [27], also provided a generic adversary in a general model that encompasses a large class of ACNs. That work, however, concentrated on the bandwidth overhead in terms of dummy messages for protocols based on pool mixes specifically. Their result does not give insights into the relationship between the dummy message rate, the latency overhead, the compromisation rate, and the degree of anonymity.

Recently, Ando et al. [2] derived necessary constraints for communication complexity and the degree of anonymity in the presence of active adversaries for mix-nets. That work does not capture bandwidth overhead and, more importantly, does not provide necessary constraints for protocols with user coordination. More recently, Ando et al. [3] proved about mix-nets that anonymity can only be achieved if each client transmits on average a superlogarithmic number of packets.

For other lines of work on upper bounds on anonymity for specific protocols and on provable anonymity guarantees, we refer the readers to [12].

2 Overview

In this work we present an impossibility result for sender anonymity of ACNs that allow messages to be sent, to mix, and to confuse a potential adversary with dummy messages. We measure sender anonymity based on the AnoA framework [25] as the inability of an adversary to distinguish between two different senders of their own choosing, say, Alice and Bob. We start by showing, with an intuitive counter-example, why the existing anonymity trilemma by Das et al. does not sufficiently capture this space of protocols.

Imagine a protocol in which users communicate out-of-band to initialize secret-sharing for their messages, e.g., they use a technique leveraged by DC-nets [17] with pre-setup key agreement, where each user only needs to publish their local messages.

Whenever a recipient receives a set of messages that belong together, the recipient has to combine all of them to extract the real message. There is a certain chance that when Alice sends her message, Bob is one of the users who provides a share. In this instance, no matter the level of compromisation or the latency overhead of the protocol, the adversary won't be able to know who out of Alice or Bob actually initiated the message.

This property was not captured in the anonymity trilemma; consequently, we may ask whether such techniques give us hope for cheap strong anonymity. We now set out to formally show that this hope is unfounded (or, at the very least, requires stronger techniques than currently available).

We use the term shares to refer to messages created to confuse the adversary in such a way, and use the term user coordination to refer to the process. By these terms we do not refer to specific techniques, but rather capture all sorts of techniques that lead to this effect.

2.1 How we prove impossibility

To show that there is not (and, in fact, cannot be) a protocol that provides strong anonymity without a significant bandwidth overhead and/or latency overhead, we need to capture all possible protocols and show that each of them is vulnerable to attacks.

Impossibility proofs. The standard proof technique for such a result is to, first, specify formally which protocols are considered; second, to come up with a specific adversary against such protocols; third, to show that there is some idealized protocol that is at least as good as any other protocol in defending against this adversary; and finally, that even this idealized protocol is vulnerable to that specific adversary. These steps let us conclude that any protocol (which cannot be better than the idealized one) is vulnerable to this specific adversary. Our specific adversary is fairly simple and possibilistic. There are more sophisticated adversaries that, e.g., take the expected distribution for each sender into account. To such adversaries the protocols are potentially even more vulnerable; hence, our results might be untight.

The most important step is to be specific and careful in the definitions of how users send messages, how protocols can send them from one party to the other and what exactly constitutes anonymity.

User message distributions. The user message distribution describes how we select which user sends messages at which point in time. This is crucial for anonymity as it defines which users are active and how often they participate in the protocol. Note that the user message distribution does not restrict an online user from sending dummy messages or shares; the user message distribution does, however, decide when and how often users want to send real messages.

As we strive to provide a result comparable to the anonymity trilemma [12] we consider two types of user message distributions: a *synchronized* user message distribution in which users take turns for sending messages. This distribution is fairly protocol-friendly, as the protocol has no uncertainty about the number of messages available at any time and can choose its strategy accordingly and in a static way. The second user message distribution we consider is the *unsynchronized* user message distribution, where each round each user independently at random decides whether or not to send a (real) message this round. This distribution creates more realistic patterns of behavior and it introduces an uncertainty for the protocol that now might want to apply dynamic strategies. We refer to Section 3.5 for more details on user message distributions.

Protocols. For showing an impossibility result, we define protocols via the constraints that we impose on them. These are as follows: protocols operate in synchronous rounds; for any packet to be sent from one protocol party to another, the sending of said packet is observable in the network; in every round, every packet can only traverse one "hop" (from one party to one other), i.e., packets cannot be sent onward in the same round they were received; when a packet is transported from one party to another the adversary can observe that fact but not the content of the packet, and packets cannot be merged. We prominently restrict protocols with two overheads: a *latency overhead* ℓ, describing how many rounds may pass between a user wanting to send a message and the correct recipient actually receiving the message, and a *bandwidth overhead* B, describing how many dummy messages or shares the protocol is allowed to use for every real message. Sections 3.2 and 3.6 discuss the protocol model in detail.

AnoA, strong sender anonymity and impossibility results. We allow our adversary to eavesdrop on the whole network (*global passive adversary*) and to additionally passively compromise a number of c out of K internal protocol parties. We measure the success or failure of the adversary in de-anonymizing users via computational indistinguishability in the AnoA framework: the adversary chooses two users as possible senders for a message to a specific adversarial recipient; after observing communication between protocol parties, the adversary guesses which of the two users actually has sent the message. We quantify the adversarial advantage δ as the improvement of the adversary's guess over a purely random guess.

Given a security parameter η that we can explicitly relate to protocol parameters such as the total number of users N, the number of parties K, or the overhead parameters for latency ℓ or bandwidth B, we say that a protocol provides *strong sender anonymity* if the adversary's advantage δ is asymptotically constrained by a negligible function (in the security parameter η).

Both the notions of strong anonymity and the definition of the AnoA framework allow the adversary a significant amount of knowledge and leverage, which is bound to weaken impossibility results based on the resulting anonymity notion. We think, however, that this choice is still meaningful and reasonable: Strong anonymity describes that the protocol is guaranteed to provide meaningful anonymity under repeated observation. A protocol cannot satisfy strong anonymity if repeated observation would compromise anonymity.

From a technical point of view, strong anonymity allows us to avoid discussions about a specific cutoff point for the adversary in favor of an asymptotic view while forcing protocols to utilize interesting strategies for providing anonymity.

The AnoA framework assumes that the adversary has an arbitrary degree of background knowledge, including knowledge of the two potential users who might send the challenge message. Note that as long as *strong sender anonymity* is considered, this notion is equivalent to having the adversary choose from an up to polynomially larger set of potential senders.

We acknowledge that in many practical cases one may choose a weaker anonymity notion that, e.g., restricts the advantage to 1/poly for some polynomial. Our results reason about such cases as well: While Theorems 2, 3 and 5 to 8 rely upon the notion of strong anonymity, the more technical Theorems 1 and 4 can be used to derive results for weaker notions as well.

2.2 Lessons learned

As we are trying to steer the research community into the right direction, we need to look at important protocol techniques. Protocols like DC nets and Dissent show that coordinating messages between several users can achieve a fundamentally different (and stronger) variant of anonymity than mixing independently generated messages. We show that, even if user coordination is utilized and implemented efficiently, it does not invalidate the general premise of the trilemma for AC protocols. Technically, we achieve this by abstracting away protocol features via this abstract user coordination.

Our analysis gives lower bounds for the adversarial advantage, and, as part of the proof technique, introduces design goals for a novel ideal protocol that might be of independent interest. As the bounds we derive are, admittedly, somewhat technically involved and hard to intuitively understand, we boil down the formal insights to a few simplified lessons formally learned from them. The goal of these lessons is not to arrive at astounding new conclusions but to formally show that existing intuitions are correct.

If the bandwidth overhead used for shares is excessive (for every message of every user, every other user sends a (share) packet), strong anonymity is possible (see DC-nets) – for any values of ℓ or c.

If the bandwidth overhead is not excessive (less than all other user sends a share) we derive more fine-grained insights from our formulas, depending on the number c of compromised nodes and the latency overhead ℓ. Most of them can be expressed in the form: strong anonymity is impossible if the product of the latency overhead ℓ and the message rate p (a measure that combines real messages and shares) is smaller than some value X.

- Even without compromised nodes ($\mathsf{c} = 0$), strong anonymity is impossible if $\ell p < 1$ (this universal constraint is from [12]; we show that it still holds under user coordination).
- If $\mathsf{c} > 0$ then strong anonymity is impossible if the latency overhead is constant ($\ell \in O(1)$).
- If at least half of all parties are compromised ($\mathsf{c}/\mathsf{K} \geq 1/2$) and $\ell \leq \log(\eta)$, strong anonymity is impossible if $\ell p \in O(1)$; ℓp must grow with η.
- Under the *Anytrust* assumption (and similar assumptions where only a constant number of parties is honest, $\mathsf{K} - \mathsf{c} \in O(1)$), we show even tighter constraints. Unless the latency grows at least quadratically with the security parameter, ℓp needs to grow more than logarithmically to provide strong anonymity (more specifically, strong anonymity is impossible if $\ell < \eta^2$ and $\ell p < \sqrt[4]{\eta}$).

Beyond these insights, we stress that for recipient anonymity shares do not help, since recipient anonymity is defined as a property of tracking individual packets. While shares can obfuscate the real sender among several users, they do not prevent the tracking of any individual packet to a recipient. Therefore, recipient anonymity bounds from [12] do not change with user coordination, and in this paper we focus on the scope of improvements of sender anonymity.

ℓ	Latency overhead for every message
β	number of noise packets for every user per round
B	number of noise packets per real message
p	Probability to send a message per user per round
p'	Probability to send a real message per user per round
K	Number of (internal) protocol parties
c	Number of compromised protocol parties
N	Number of online users (that may send messages)
δ	Adversarial advantage in the anonymity game
Π	A protocol. $\Pi \in M$: Π is within our model
η	The security parameter
ϵ	A (very small, but non-negligible) function

Fig. 1. Notation, as in Das et al. [12]

3 Preliminaries

We use the same notation as Das et al. [12]; see Figure 1 for a full notation table.

3.1 Anonymity definition

We define (sender) anonymity by a game between a challenger (controlling the protocol) and a global passive adversary, following the AnoA framework [25]. The challenger receives all protocol parameters and a description of how users want to send messages (the user distribution), as well as a challenge bit b that influences which of two adversarially chosen senders actually sends a particular challenge message. The adversary's goal is to guess this challenge bit based on its observations. In this section we briefly introduce the relevant concepts of this anonymity game. We start with sender anonymity.

Consider an interactive game $\langle \mathcal{A} | \mathtt{Ch}(\Pi, b) \rangle$ between a challenger and the adversary, where the adversary can send messages of two flavors:

- $(\mathtt{Input}, u, R, m)$, which prompts the challenger to make user u send a message m to recipient R.
- $(\mathtt{Chall}, u_0, u_1, R, m)$, in which case the challenger selects one user based on the challenge bit b, and then instructs user u_b to send a message m to recipient R.

After receiving the adversarial inputs, the challenger $\mathtt{Ch}$ runs the protocol Π based on these choices. $\mathtt{Ch}$ then forwards all adversarial observations to $\mathcal{A}$.

Definition 1 (δ-sender anonymity from [12])**.** *A protocol Π provides δ-sender anonymity for a class of adversaries C and a function $\delta(\cdot) \geq 0$ describing the maximal adversarial advantage, if for all PPT machines $\mathcal{A} \in$ C,* $\Pr[0 = \langle \mathcal{A} | \mathtt{Ch}(\Pi, 0) \rangle] \leq \Pr[0 = \langle \mathcal{A} | \mathtt{Ch}(\Pi, 1) \rangle] + \delta(\eta)$.

Π provides strong sender-anonymity *[12, 15, 18] if it provides δ-sender anonymity for a function $\delta(\eta)$ that is negligible in the security parameter η.*

On the meaning of η. In our analyses we tie η to system parameters such as the number of parties K, the number of compromised parties c, the latency overhead ℓ, the bandwidth overhead B, the number of users N, etc.; we explicitly describe the relationship between η and these parameters for the cases we consider. The system parameters don't have to increase with η necessarily. In some cases, parameters may decrease as η increases, for example, the bandwidth overhead B might decrease as the latency overhead increases, or the ratio of compromised (or honest!) parties might decrease.

Note that if an AC protocol has strong anonymity, it is secure under continual observation (e.g., for streams of messages or usage over a longer time period) and formally, η limits the number of observations.

On anonymity sets and strong anonymity. Strong sender anonymity lets the adversary freely choose the pair of challenge senders and requires that the AC protocol's behavior is indistinguishable to the adversary. Hence, strong sender anonymity corresponds to the full anonymity set (see [25, Lemma 7]) that encompasses all users. Sender anonymity for (non-full) anonymity sets would result in restricting the adversary in Definition 1 to choose the pair of challenger senders from the same anonymity set.

3.2 What can and cannot protocols do?

Anonymous communication protocols are communication protocols, so we require them to ultimately transmit messages from senders to recipients; these messages are encoded in packets of information. A protocol may utilize its set of (internal) protocol parties P to mix, delay or modify packets (i.e., encrypt or decrypt them).

Time. We use a round-based definition of time in which we assume that all protocol parties work in synchronized rounds. In each round, a party can send packets to other parties that will receive the packets at the end of the round (and can then send them on in the next round). We allow, but abstract away from any cryptographic operations locally performed on these packets and we don't consider the computation time required for such operations: independently of the cryptographic operations performed, a packet is always ready for being sent in the round after it arrived.

We define the latency overhead ℓ of a protocol as the number of rounds that pass between the round in which a message is scheduled for being (originally) sent by a user u and the round it is received (and potentially reconstructed) by a recipient R. We define the *bandwidth overhead* B as the number of noise messages that the protocol can create for every real message.

Finally, we allow the protocol to leverage what we call user coordination. This term doesn't specify any one technique as much as it is an umbrella for a wide range of strategies that leverage cooperation among users to improve anonymity. The next subsection expands on the intuition and techniques behind user coordination.

3.3 User coordination

In this subsection we introduce a concept fundamental to our analysis: User coordination (UC) allows protocols to leverage coordination between clients to add a layer of uncertainty on top of what they could achieve by mixing of messages.

Before discussing how we formally define UC, we want to clarify a few important points about the concept: First and foremost, UC is not a single technique, but a sound over-approximation of different techniques that protocols could employ. These techniques include secret-sharing of messages, sending encryptions of 0's (e.g., in some DC-nets) and threshold encryption. Our focus here is not on how exactly specific techniques under the umbrella of user coordination can be implemented, but rather on showing that even such strong techniques do not allow protocols to overcome a fundamental anonymity trilemma. The following definition captures, in an abstract manner, what benefits a protocol could achieve when employing UC.

Definition 2 (User Coordination)**.** *In a protocol with* user coordination *(UC) users can send packets to help anonymize a message m sent by a user u. If h users other than u send their packets, then the recipient R of m can retrieve m if and only if R has access to all $h+1$ packets (which we call* the shares of *m).*

Even a malicious recipient cannot determine which of the $h+1$ shares contained m. This feature effectively provides the sender of a message with the ability to hide among a group of $h+1$ users.

Even though we use the term *share*, we do not imply that the message needs to be secret shared (as in cryptographic secret sharing). Our over-approximation about UC is sound because, if less than h+1 shares suffice to reconstruct a message, the recipient can only learn more about the potential senders (e.g., by being able to exclude some shares the adversary can exclude some potential senders).

Costs and complexity of UC. UC is an over-approximation of a wide range of strategies a protocol could employ to confuse an adversary. With our paper we show that protocols that employ any such techniques are still subject to an anonymity trilemma. Our model soundly abstracts away from the potential overhead that UC incurs: we allow the preparation for UC to happen out of band, i.e., we consider it to be "free". [1] However, h out of the $h+1$ shares are considered overhead messages for the purpose of determining bandwidth overhead. We will see that choosing $h = B$ is optimal.

3.4 Adversary

Following [12], we consider global passive adversaries, that observe all communication between protocol parties and that can additionally compromise c protocol parties. These *compromised parties* still follow the protocol specification and thus are considered honest but curious or *passively compromised.*

We assume that our adversary does not or cannot interfere with packets in transmission and cannot link packets sent by a party to packets previously received by that party, except if the party is compromised. This is equivalent to assuming an authenticated and encrypted channel between all parties.

No active attacks. It is worth repeating that we strive for impossibility results. While the adversary would nat-

[1] Allowing UC to be set up essentially for free does not mean we expect this to be particularly easy for every protocol. Rather, our necessary constraints even hold if someone managed to implement UC without any additional overheads.

urally be more powerful when performing active attacks, this restriction strengthens our results: even without active attacks, we show upper bounds on anonymity.

3.5 User message distributions

We follow [12] in their distinction between two types of *user distributions*, i.e., two different definitions of how users interact with the protocol; Das et al. distinguish between a *synchronized* user distribution U_B and an *unsynchronized* user distribution U_P. In the synchronized user distribution U_B the users (globally) agree that every round exactly one user gets to send a message, while other users may or may not send noise messages (within the bandwidth overhead). In the unsynchronized user distribution U_P every user flips a (biased) coin with success probability p in every round, independently of other users, to determine whether or not they will send a message (real or noise) in this round.

The synchronized user distribution can be seen as a control group that is predictable and thus fairly protocol friendly. Protocols following DC-nets tend to use such a synchronization (to ensure that messages from a sender can actually be reconstructed). Our results show that many interesting cases are the same for this predictable user distribution U_B and for the unsynchronized U_P.

Difference between synchronization and UC. Note the difference between *synchronization* of our user message distribution and user coordination. The distributions describe shapes of user traffic, i.e., users decide *when* to send their own messages; whereas what we call user coordination is a powerful way in which users can coordinate their message content to confuse the adversary, i.e., it tells each of the $h+1$ users *what* to send so that the combination of the $h+1$ packets will allow the recipient to obtain the original message.

3.6 Protocol model for ACNs

We follow [12] in our definition of a protocol model but extend a protocol's capabilities by allowing user coordination. Protocols in our model work as follows:

- a set of senders $\mathcal{S}$ can send packets to a set of recipients $\mathcal{R}$ via some anonymizing proxies P.
- Protocols operate in rounds. In every round, the protocol can decide to keep a packet with the same protocol party, or move the packet to a different protocol party, as long as the latency overhead constraint is not violated. No packets can be dropped by the protocol.
- Whenever a packet is sent from one party to another, the eavesdropping adversary learns that a packet is sent as well as the round in which this occurred.
- Additionally, the adversary is allowed to compromise a number of c proxies. Whenever one of these compromised proxies sends a packet, the adversary learns to which (previous) incoming packet it corresponds; otherwise the adversary does not learn this.
- Besides, the adversary compromises all recipients and upon receiving packets can learn their content.
- However, UC techniques add to the adversary's confusion here by requiring receiving several packets to (re-)construct a real message. We call all packets required to reconstruct a message m the *shares* of m and we identify them by assigning them the same tag (some randomly generated string). The tag of a packet is revealed only to the (final) recipient of a packet.

We do not restrict our protocols to any specific technique of UC; it can be any method (e.g., secret sharing, multi-party computation etc.) that achieve the user coordination property. However, for our impossibility analysis, we assume that user coordination can be achieved via a pre-processing step or can be done efficiently, and hence, we ignore the cost of user coordination. For completeness, we describe the formal protocol model along with changes from [12] in Appendix B.

Note that we do not consider any compromised sender in $\mathcal{S}$ for our analysis. If $\mathsf{N_c}$ out of N senders are compromised, the adversary can always correctly identify their packets at the recipient, and hence, they don't contribute towards anonymity. This is equivalent to analyzing $\mathsf{N} - \mathsf{N_c}$ honest users and therefore, for simplicity, we assume all users are honest.

Assumptions on user coordination. We impose the following assumptions on user coordination:

1. If $h+1$ shares are used to reconstruct a message, at least one of them is sent by the original sender.
2. A packet cannot have multiple tags, i.e., no share can take part in reconstructing two separate messages.
3. A compromised protocol party is always able to map its outgoing packets to its incoming packets.

Our assumptions are consistent with most ACNs from the literature. In Appendix B.3, we discuss how these assumptions present cryptographic challenges.

Additionally, to be consistent with how we count the latency overhead for a real message, we add the restriction that every packet (real or noise, created by a user or an internal protocol party) is allowed to remain in the system for no more than ℓ rounds. Finally, if a message is scheduled to be sent in round t_0 by the user distribution, all shares of that message (as well as the real packer) must reach the recipient before round $t_0+\ell$.

Expressiveness of protocol model M. Unless excluded by the above assumptions, our protocol model M can be used to express any protocol. Our model is an extension of the one from [12]; therefore, any protocol techniques that can be expressed in their model, can be expressed in our model. Mix networks very naturally fit into their model. Other than that, protocols based on Private Information Retrieval, Multi-party Computation, peer-to-peer protocols can be captured in their model, as long as there is no user coordination. By extending the model for protocols with tags we capture user coordination. The only protocol behaviors that can not be expressed by our model are (1) the exploitation of side channels, (2) using unreliability or worst-case infinite latency for anonymity, and (3) techniques that allow mixing in dishonest nodes. We refer to Appendix B.2 for a discussion of these limitations.

4 Towards a new trilemma

We investigate the fundamental limitations of protocols. To this end, we define an abstract protocol within our model that leverages user coordination combined with mixing techniques. We then show that this protocol can achieve a better degree of anonymity than the classical impossibility results of Das et al. indicate.

The intuitive reason for this effect is that such a protocol introduces an additional form of uncertainty for the adversary that was not captured by the classical impossibility results. Imagine an adversary that compromises every node in the path that a particular packet traverses and then observes that the packet is being used to reconstruct a message. This adversary might not always learn who actually sent the reconstructed message: all the packets with shares that belong together have to be combined to learn the message; thus all potential senders of these packets could be the message's sender.

We then show an anonymity trilemma that captures even protocols with user coordination: every protocol in our model can be defeated by a straight-forward path adversary unless the protocol utilizes sufficient bandwidth and latency overhead that depends on the degree of compromisation in the network.

4.1 AC leveraging user coordination

We now describe a protocol that falls within our protocol model (Section 3.6) and that leverages user coordination to provide more anonymity than postulated in the impossibility results of Das et al. [12] for some values of ℓ, β, and c. While this doesn't show that their result is wrong (they didn't consider user coordination), it emphasizes the importance of covering such protocols in an impossibility result.

The main idea that allows this to work is that we use our bandwidth overhead for shares. Each such share is associated with one real message (with content) within the system and the recipient needs to collect all the shares of a message to decipher it. When all the shares of a message reach a recipient, the adversary can thus only learn that the message has reached and which packets were involved in reconstructing it, but not point to one specific packet it was in.

We assume that the adversary can not break the sharing of message origin provided by user coordination and hence can not decipher an individual message before it reaches the recipient. Additionally, we assume that our user coordination happens out-of-band and is efficient. (For instance, in DC-net [17] with pre-setup key agreement, the protocol parties only need to publish their local messages.)

The protocol works in the following way:

1. Users send messages based on a given user message distribution (c.f., Section 3.5).

2. All users participate in the out-of-band user coordination. Instead of sending a dummy noise message, users send shares for other users' messages.

3. Users run an out-of-band consensus protocol to decide when their messages (real message or share) will be delivered, such that in a given round the recipient receives shares of the same message and all the shares of that message (comparable to t-out-of-t secret sharing).

4. In a given round, the recipient combines all the shares that he receives to extract the real message.

5. The protocol utilizes a series of up to K relays; as long as messages (real or share) are in the system, they are sent from one relay to the next. Note the attacker can compromise up to c of these relays. To prevent the attacker from compromising a consecutive series of relays, we permute the order in which relays are being used. Once the protocol starts, the sequence of the relays is sent to all users.

Analysis of adversarial advantage for the above protocol. We know from Das et al. (c.f. Appendix C)

that for the *synchronized user distribution*, the adversarial advantage δ should be lower bounded by
$\delta \geq 1 - \left[1 - \binom{\mathsf{c}}{\ell}/\binom{\mathsf{K}}{\ell}\right] \times min\left(1, \frac{\ell+B\ell}{\mathsf{N}-1}\right)$.

Recall that according to the anonymity notion (c.f., Section 3.1), the adversary has to distinguish between two potential senders of a message, u_0 and u_1. If, say, u_0 sends the challenge message and the adversary has compromised every entity on the path this message takes from u_0 to the recipient, then the classical trilemma insists that the adversary wins, which is correct for protocols without user coordination. With user coordination, however, it is possible that u_1 sends a share of the challenge message. This occurs with probability $\frac{B}{\mathsf{N}-1}$. If this happens, there is no way in which the adversary can know whether u_0 or u_1 has sent the challenge message (even if the whole path was compromised) and hence $\delta \leq 1 - \frac{B}{\mathsf{N}-1}$ must hold. We directly see a conflict between the anonymity achieved by our protocol and the impossibility result. For an illustrative example consider the case where $\ell = 1, \mathsf{K} = 2, \mathsf{c} = 1$. We compare the upper bound on δ derived directly from user coordination with the lower bound from [12] and yield:

$$1 - \frac{B}{\mathsf{N}-1} < 1 - \left[1 - \binom{\mathsf{c}}{\ell}/\binom{\mathsf{K}}{\ell}\right] \times min\left(1, \frac{\ell+B\ell}{\mathsf{N}-1}\right)$$

$$\Longleftarrow \frac{B}{\mathsf{N}-1} > \frac{1}{2} \times \frac{\ell+B\ell}{\mathsf{N}-1} \qquad \text{assuming } \frac{\ell+B\ell}{\mathsf{N}-1} < 1$$

$$\iff 2B > 1 + B \iff B > 1.$$

Thus, with just one noise message per real message ($B = 1$), our protocol violates the classical impossibility bounds. More generally, if a set of users sends shares for a given message, the adversary can not distinguish the actual sender of the message from other users in the set, unless the user coordination is broken. Note that this effect is similar to the amount of uncertainty introduced by messages meeting (and mixing) in an honest relay.

4.2 The path possibility adversary

Having shown a separation, we now build up to our own impossibility result: a new lower bound on the adversarial advantage (upper bound on anonymity) in the presence of compromised relay nodes, both for synchronized users and unsynchronized users. To this end, we first introduce the path adversary, then present a necessary invariant for anonymity and finally present an ideal protocol that, although provably superior to all protocols within our model, still falls prey to our adversary.

Formally, we utilize a path possibility adversary $\mathcal{A}_{paths}$ as in the work of Das et al.[12]: The adversary observes all communication patterns of all users. Upon arrival of the challenge message at the recipient, the adversary checks whether one of the challenge users could not have sent this message, i.e., whether it is impossible to construct a consecutive path from the user to the challenge message's arrival that satisfies the latency constraint. If one of the users can be excluded in this way, the adversary guesses that the other user sent the challenge message. Otherwise, the adversary simply flips a coin to decide which challenge user to output. For a complete description of the adversary see Appendix B.4.

4.3 Necessary invariant for anonymity

To prove their anonymity trilemma for protocols without user coordination, Das et al. defined a necessary invariant, i.e., if the invariant is not satisfied for a protocol run, then even the path possibility adversary $\mathcal{A}_{paths}$ will certainly win independently of any specific actions or strategies utilized by the protocol.

Invariant 1 (Necessary invariant from [12]). *Let u_0 and u_1 be the challenge users; let b be the challenge bit. Assume that the challenge message reaches the recipient at r. Assume furthermore that u_{1-b} sends her messages (including noise messages) at $V = \{t_1, t_2, t_3, \dots\}$. If $T = \{t : t \in V \wedge (r - \ell) \leq t < r\}$,*

(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, $t' \in \{\min(T), \dots, r-1\}$.

Critically, this invariant *is not necessary* for protocols with user coordination (see our example above in Section 4.1). We now derive a new invariant that remains necessary for anonymity in the presence of protocols with user coordination.

Invariant 2 (New Invariant). *Let u_0 and u_1 be the challenge users; let b be the challenge bit. Assume that the challenge message reaches the recipient at time r. Assume furthermore that u_{1-b} sends her messages (including noise messages) at $V = \{t_1, t_2, t_3, \dots\}$. If $T = \{t : t \in V \wedge (r - \ell) \leq t < r\}$,*

(i) the set T is not empty, AND

(ii) (a) at least one share of the challenge message is dispatched by u_{1-b} in $\{(r-\ell), \dots, (r-1)\}$, OR
(b) at least one share of the challenge message passes through an honest node at time t' such that $t' \in \{min(T), (r-1)\}$, AND at least one of the messages (real message or noise) from u_{1-b}, sent at

$t \in \{(r-\ell), \ldots, (r-1)\}$, passes through an honest node at time t' such that $t' < r$.

Claim 1 (Invariant 2 is necessary for anonymity)**.** *Let Π be any protocol $\in M$ with latency overhead ℓ and bandwidth overhead B. Let u_0, u_1, b and T be defined as in Invariant 2. If Invariant 2 is not satisfied by Π, then our adversary $\mathcal{A}_{paths}$ as in Section 4.2 wins.*

We refer to Appendix D for the proof.

Intuition about Invariant 2 and Claim 1. The invariant establishes minimal conditions for anonymity to hold against a path possibility adversary. To this end, we look at which cases would allow the adversary to defeat the protocol and in which cases the adversary can be fooled. Note that the adversary knows the two potential challenge users and can observe the traffic, but can only connect incoming and outgoing messages of a compromised party. The adversary can also see when the challenge message reaches the recipient.

- If only one of the two challenge users sends a message in the ℓ rounds before the challenge message reaches the recipient, then only that challenge user could have sent the message without violating the latency constraint. This observation is captured in part (i).
- If both challenge users happen to collaborate on sending the challenge message (u_b is the actual sender of the challenge message, u_{1-b} happens to send a share for this specific message), then the adversary cannot decide which of the two users has sent the challenge message. Even a more realistic adversary would, in most cases, lose this game. The only way to still decipher which user sent the message is to exploit other information (say, about other messages sent by the two users), but the path possibility adversary does not attempt this. This observation is captured in part (ii a).
- In case (ii a) does not occur, there are two other cases in which the path possibility adversary wins: (1) if the adversary can track all the shares of the challenge message from sender to recipient (since we assume (ii a) does not hold, these senders don't include u_{1-b}); (2) if the adversary can track all the packets sent by u_{1-b} to their respective recipients and thus be sure that u_{1-b} has not sent the challenge message. We see that in both of these cases the path possibility adversary wins. Thus, they have to be necessary for anonymity. This observation is captured in part (ii b).

Overall, Invariant 2 describes the following logical formula: (i) AND (ii), where

- (ii) = (ii a) OR (ii b)
- (ii b) = (ii b 1) AND (ii b 2)

Lessons for an ideal protocol. Let us now look at the parts of Invariant 2 and discuss what they mean for constructing an ideal protocol:

- (i) The set T must not empty. This depends solely on when users send (real or noise) messages, which we capture with our definitions of the user message distribution. Thus, this part is independent of the decisions of the actual protocol.
- (ii a) The user u_{1-b} sends a share of the challenge message. To maximize this probability, which is independent of our other choices, we want the chance that any user is sending shares for any other user to be as large as possible.
- (ii b 1) At least one share of the challenge message travels through an honest node. To maximize the probability that this occurs, a protocol should maximize the number of nodes collectively visited by shares.
- (ii b 2) At least one message from u_{1-b} travels through an honest node. Since the (ideal) protocol doesn't know which users are the challenge users it needs to generalize: To maximize the probability that this occurs, the ideal protocol should maximize the number of nodes collectively visited by messages *from each user*.

These lessons, particularly (ii b), inspire our choices for an ideal protocol. Before we explore them, we briefly discuss the role of internal noise messages and relate them to the invariant.

4.4 Modeling internal noise

To make the accounting of bandwidth overhead easier we want to disallow the protocol from using internal noise, i.e., noise packets generated by a protocol party $\notin \mathcal{S}$. Recall the assumptions we place on all packets, including internal noise: 1. No packets can be dropped. 2. Packets can be tagged as a share of message m, but only with one tag and that tag can never be changed. 3. Packets can remain in the system for at most ℓ rounds from their generation. 4. Shares must not violate the latency bound of the message that the noise is tagged with (c.f. Section 3.6); i.e., for a message m, all packets tagged with m must arrive within ℓ rounds of the round in which the user wanted to send m.

Claim 2 (User noise can replace internal noise)**.** *For every protocol in which noise is generated by internal protocol parties ($\notin \mathcal{S}$) and latency overhead ℓ, there exists a protocol that uses only user-generated noise (noise packets originating from a user $u \in \mathcal{S}$) and la-*

tency overhead $\ell + 1$ with at least equal probability of satisfying Invariant 2.

If a noise message is generated by an internal party P we randomly choose a user to generate it and then relay it to P. We refer to Appendix D for the proof.

4.5 Ideal protocol

We now construct a protocol Π_{ideal} that has a probability of satisfying Invariant 2 against $\mathcal{A}_{paths}$ at least as high as any other protocol in our protocol model.

Following Claim 2, we allow our ideal protocol to have latency overhead of $\hat{\ell} = \ell{+}1$, and assume that every message is created by some user $u \in \mathcal{S}$. Consequently the adversary behaves as if he is dealing with a protocol that is allowed to have $\hat{\ell}$ latency overhead.

The protocol has a number of pre-defined paths. Those paths are constructed at the beginning of the protocol and do not change throughout the protocol run. Π_{ideal} has access to an oracle O (discussed later); Π_{ideal} calls O.QueryPaths() to decide the number of paths and distribution of protocol parties in each path. Fig. 2 presents pseudocode for the ideal protocol.

Since the protocol has control over the noise messages, it utilizes all the noise messages as shares of some real message. Whenever a message (real or noise) is sent to a path Path it is sent to the protocol party at position $r \mod |\mathsf{Path}|$ in the path, if the current round number is r. In the next round either the message is delivered to the recipient, or transferred to the next protocol party (at position $(r+1) \mod |\mathsf{Path}|$) in the same path. For every message m, Π_{ideal} queries the oracle (by calling O.QueryForMessage(m)) to decide which path the message should be sent to and the number of rounds the message should remain in the protocol. If the message is a noise message, the oracle additionally returns the real message that the noise should be a share of.

The oracle O is an overapproximation of different strategies that a protocol can use to optimize paths and noise messages. Our oracle knows the user distribution, all past and future messages, the number of compromised parties, and the protocol strategy. The protocol is oblivious to the challenge message, the challenge bit, the challenge users, the identity of the protocol parties who are compromised; and so is the oracle. Thus, given the user distribution, the past and future messages, and the number of compromised parties, the oracle tries to maximize the probability of satisfying Invariant 2 for the protocol Π_{ideal}, against our adversary $\mathcal{A}_{paths}$.

The oracle O can achieve the above by trying out all possible configurations and calculating for each configuration the probability of satisfying Invariant 2 assuming that the two challenge users are chosen uniformly at random (refer to Fig. 4 in Appendix B.5 for a possible instantiation of the oracle.). This consideration is different from the protocol actually satisfying the invariant, since the oracle does not actually run the protocol; is unaware of the actual challenge users, the exact protocol parties that are compromised, or the actual challenge message. Note that the oracle is not bounded polynomially anymore; however, since we are proving impossibility, a stronger protocol still provides a valid impossibility result.

```
Oracle O;
Paths ← O.QueryPaths();
MessageRoute⟨m, path, delay, tag⟩ ← empty set;

Upon Round $r$:
  for each Path in Paths do
    $i \leftarrow r \mod |\mathsf{Path}|$
    for each message $m$ held by party Path[$i-1$] do
      (path, delay, tag) ← MessageRoute.Get($m$)
      if delay is expired then send $m$ to recipient
      else send $m$ to Path[$i \mod |\mathsf{Path}|$]
  for each message $m$ in the input queue do
    (path, delay, tag) ← O.QueryForMessage($m$)
    MessageRoute.Add(⟨m, path, delay, tag⟩)
    Path ← Paths[path]; send $m$ to Path[$r \mod |\mathsf{Path}|$]
```

Fig. 2. Definition of Ideal Protocol Π_{ideal}

Special case of Ideal Protocol when $\mathsf{K} > (B+1)\hat{\ell}$. For a special case of parameters, we can construct a fairly practical ideal protocol that does not require as much help from the oracle. Consider the case where the number of protocol parties K is large enough so all shares of a message can travel on distinct paths that do not overlap; also assume for simplicity that we have a constant rate of input messages per round. In this case, we can use static looping paths, where each path is comparable to the ideal protocol from Das et al.: packets on each path remain together and hop from one node to the next (on that path).

Technically, we define $B+1$ paths with an approximately equal number $(K/(B+1))$ of mutually exclusive protocol parties each. Whenever a user sends a packet, the protocol queries the oracle for the latency and the path index, but the paths themselves remain the same. Otherwise the ideal protocol remains unmodified.

The ideal protocol is ideal. We now show that the ideal protocol is indeed ideal for Invariant 2.

Claim 3 (Ideal protocol is ideal for Invariant 2)**.** *Against the given adversary $\mathcal{A}_{paths}$, Π_{ideal} with latency $\hat{\ell}$ satisfies Invariant 2 with probability at least as high as any other protocol in M with latency ℓ.*

We refer to Appendix D for the proof. From here onwards, we assume that messages (real or noise) are generated only by users $\in \mathcal{S}$, and whenever a latency of ℓ is allowed to the protocol, we allow the ideal protocol to have a latency of $\hat{\ell} = \ell + 1$ in our calculations.

5 Analyzing synchronized users

The first user distribution we analyze is the synchronized user message distribution U_B as defined in Section 3.5. Recall that in U_B in every round exactly one user sends a message and within N rounds each user sends a message only once. We allow the protocol to add up to B noise packets per round and leave it up to the protocol to decide which users send those B packets. If B is not a natural number, we allow the protocol to send $\lfloor B \rfloor$ noise messages per round and one more every few rounds s.t. the average bandwidth overhead remains B while spacing them out as evenly as possible.

5.1 Lower bound on adversarial advantage

Theorem 1. *For user distribution U_B, no protocol $\Pi \in M$ can provide δ-sender anonymity, for any*
$\delta < \left(1 - \frac{B}{\mathsf{N}-1}\right) \left[1 - \frac{(\tau+1)\mathsf{N} - B\hat{\ell} - \hat{\ell}}{\mathsf{N}} g(\tau) - \frac{B\hat{\ell}+\hat{\ell}-\tau\mathsf{N}}{\mathsf{N}} g(\tau+1)\right]$
where $\tau = \lfloor \frac{B\hat{\ell}+\hat{\ell}}{\mathsf{N}} \rfloor$, $\hat{\ell} = \ell + 1$
and
$$g(x) = \begin{cases} 1 & \mathsf{c} < x\hat{\ell} \\ 1 - \binom{\mathsf{c}}{x\hat{\ell}} \Big/ \binom{\mathsf{K}}{x\hat{\ell}} & \mathsf{c} \geq x\hat{\ell}. \end{cases}$$

Proof. Suppose u_0 and u_1 are the challenge users, and u_b sends the challenge message which reaches the recipient in some round r. We know from Claim 3 that Π_{ideal} is ideal; thus, we focus on Π_{ideal} here. By definition of Π_{ideal}, the challenge message can have up to $(B+1)$ shares, including the one sent by u_b. Since the challenge users are not known to the oracle O, the best strategy for O is to have B shares per real message.

For our invariant to be satisfied, it is necessary that u_{1-b} sends at least one message within $[r-\ell, r-1]$. Such a message can be a share of the challenge message, or a real message. If we denote by x the number of messages sent by u_{1-b} in a given interval of $\hat{\ell}$ rounds, x can have only two possible values depending on the values of B, $\hat{\ell}$ and N. That is because the protocol tries to maximize the total number of users that send messages in a given interval of ℓ rounds. Hence, u_{1-b} sends $\tau = \lfloor \frac{B\hat{\ell}+\hat{\ell}}{\mathsf{N}} \rfloor$ messages with probability $\frac{(\tau+1)\mathsf{N} - B\hat{\ell} - \hat{\ell}}{\mathsf{N}}$, and sends $(\tau+1)$ messages with probability $\frac{B\hat{\ell}+\hat{\ell}-\tau\mathsf{N}}{\mathsf{N}}$.

If none of them is a share of the challenge message, we require that at least one of those messages passes through an honest node before round r. Hence,

$$\begin{aligned} &\Pr[\text{Invariant 2 is true}] \\ &\leq \Pr[u_{1-b} \text{ sends a share of the challenge message}] \\ &+ \Pr[u_{1-b} \text{ sends no shares of the challenge message} \\ &\quad \wedge u_{1-b} \text{ sends a message in the given span of } \hat{\ell} \text{ rounds}] \\ &\times \Pr[\text{At least one of the messages visits an honest node}] \\ &\leq \tfrac{B}{\mathsf{N}-1} + \left(1 - \tfrac{B}{\mathsf{N}-1}\right) \tfrac{(\tau+1)\mathsf{N} - B\hat{\ell} - \hat{\ell}}{\mathsf{N}} \times g(\tau) \\ &\qquad + \left(1 - \tfrac{B}{\mathsf{N}-1}\right) \tfrac{B\hat{\ell}+\hat{\ell}-\tau\mathsf{N}}{\mathsf{N}} \times g(\tau+1). \end{aligned}$$

where $\tau = \lfloor \frac{B\hat{\ell}+\hat{\ell}}{\mathsf{N}} \rfloor$, and $g(x)$ is a function that provides an upper bound on the probability that at least one message from u_{1-b} passes through at least one honest node in a given interval of $\hat{\ell}$ rounds, when u_{1-b} sends exactly x messages. Hence,

$$\begin{aligned} &\Pr[\text{at least one message from } u_{1-b} \text{ passes through} \\ &\quad \text{an honest node } | u_{1-b} \text{ sends } x \text{ messages}] \\ &\leq g(x) = \begin{cases} 1 & \mathsf{c} < x\hat{\ell} \\ 1 - \binom{\mathsf{c}}{x\hat{\ell}} \Big/ \binom{\mathsf{K}}{x\hat{\ell}} & \mathsf{c} \geq x\hat{\ell} \end{cases} \end{aligned}$$

By Claim 1 whenever Invariant 2 is not true the adversary wins. Whenever it is true, the adversary still can flip a coin and thus the probability that the adversary loses is bounded by:

$$\begin{aligned} \Pr\left[0 = \mathcal{A}_{paths} | b = 1\right] &= \Pr\left[1 = \mathcal{A}_{paths} | b = 0\right] \\ &\leq \frac{1}{2} \Pr[\text{Invariant 2 is true}]. \end{aligned}$$

Therefore,

$$\begin{aligned} &\delta \geq 1 - \Pr[\text{Invariant 2 is true}] \\ &\geq \left(1 - \tfrac{B}{\mathsf{N}-1}\right) \left[1 - \tfrac{(\tau+1)\mathsf{N} - B\hat{\ell} - \hat{\ell}}{\mathsf{N}} g(\tau) - \tfrac{B\ell+\ell-\tau\mathsf{N}}{\mathsf{N}} g(\tau+1)\right]. \end{aligned}$$

□

Although the above bound is a perfectly valid lower bound for δ over $0 \leq \mathsf{c} \leq \mathsf{K}$, when $\mathsf{c} < \hat{\ell}$ and $\tau = 0$, we can derive a more precise lower bound on δ:

$$\delta \geq \left(1 - \tfrac{B}{\mathsf{N}-1}\right) \left(1 - \tfrac{B(\hat{\ell}-\mathsf{c}) + (\hat{\ell}-\mathsf{c})}{\mathsf{N}} - \tfrac{B\mathsf{c}+\mathsf{c}}{\mathsf{N}} \left[1 - 1 \Big/ \tbinom{\mathsf{K}}{\mathsf{c}}\right]\right).$$

Appendix F.1 presents a full derivation for this bound.

5.2 Impossibility for strong anonymity

Using Theorem 1, we can derive the following impossibility theorems. Appendix E presents the full proofs.

Theorem 2. *For user distribution U_B with $K, N \in poly(\eta)$, $K > c$, $\hat{\ell} < \mathsf{N}$, $\mathsf{N} - 1 > B \geq 0$, no protocol $\Pi \in M$ can achieve strong anonymity if*

(i) $\hat{\ell}(B+1) < \mathsf{N} - \epsilon(\eta)$ where $\epsilon(\eta) = 1/\eta^d$ for a positive constant d, OR

(ii) $c \geq (B+1)\hat{\ell}$ and $\hat{\ell} \in O(1)$.

Theorem 2 in words. For our user distribution U_B in which user behavior is synchronized and somewhat predictable, a protocol has only three options to achieve strong anonymity: (1) to use a massive amount of latency overhead $\hat{\ell} \geq \mathsf{N}$ (intuitively: "wait until everyone has sent a message"); (2) to use a massive amount of bandwidth overhead $B \geq \mathsf{N} - 1$ (intuitively: "every user sends a packet in every round"); or (3) to have a trade-off between the two: $\hat{\ell}(B+1) \geq N$ (intuitively: "make sure that messages wait long enough for everyone to have sent a packet"). In case (3) we have an additional requirement: if the adversary is allowed to compromise more than the number of parties that any message and its shares can meet while they are within the protocol, then it is possible that the challenge message and all its shares only travel through compromised nodes. If the length of the paths taken by these messages is constant, this occurs with non-negligible probability and thus strong anonymity is impossible.

Theorem 3 (Anytrust Impossibility Theorem). *For user distribution U_B with $\mathsf{K}, \mathsf{N} \in poly(\eta)$, $\mathsf{K} - \mathsf{c} = \gamma \in O(1)$, $\mathsf{K} > \mathsf{c}$, no protocol $\Pi \in M$ can achieve strong anonymity if $\hat{\ell} \leq \mathsf{N} - 1$ and $B \leq \mathsf{N} - 2$ and $\hat{\ell}^2 \leq \mathsf{K} - \gamma$.*

Theorem 3 in words. If all but a constant number of nodes are compromised, then strong anonymity is only possible with a massive latency overhead or a massive bandwidth overhead (cases (1) and (2) from Theorem 2 in words respectively) or if the latency overhead ℓ allows each packet to traverse at least the square root of all compromised parties ($\hat{\ell}^2 \geq \mathsf{c}$).

6 Analyzing unsynchronized users

We now analyze the unsynchronized user message distribution U_P as defined in Section 3.5. Recall that in U_P in every round each user tosses a biased coin with success probability $p \in (0, 1]$ to decide whether or not to send a message. This coin toss is independent of coins tossed by other users or in other rounds. We assume that the bandwidth overhead is part of p, i.e., we divide up p into the probability to send a real message $p' < p$ and define our bandwidth overhead as $B = \frac{p-p'}{p'}$ noise messages per real message.

6.1 Lower bound on adversarial advantage

Theorem 4. *For user distribution U_P, no protocol $\Pi \in M$ can provide δ-sender anonymity, for any*

$$\delta < \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\left[1 - g(\hat{\ell}) \times f_p^{SA}(\hat{\ell})\right], \textit{ where}$$

$$B_{\text{eff}} = \min(B, \hat{\ell}p\mathsf{N} - 1),$$

$$f_p^{SA}(d) = \min\left(1, \frac{1}{2} + \left(1 - (1-p)^d\right)\right), \textit{ and}$$

$$g(x) = \begin{cases} 1 - \binom{\mathsf{c}}{x\hat{\ell}}/\binom{\mathsf{K}}{x\hat{\ell}} & x\hat{\ell} \leq \mathsf{c} \leq \mathsf{K} \\ 1 & \text{otherwise.} \end{cases}$$

We refer to Appendix E for the proof of this theorem.

Although the above bound is a valid bound for $0 \leq \mathsf{c} \leq \mathsf{K}$, we can derive a more precise bound when $\mathsf{c} < \ell$:

$$\delta \geq \left(1 - \tfrac{B_{\text{eff}}}{\mathsf{N}-1}\right) \times \left(1 - f_p^{SA}(\hat{\ell} - \mathsf{c})\right) \times \left[1 - f_p^{SA}(\mathsf{c})\left(w_2 + w_1\left[1 - 1/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\right)\right],$$

where $w_1 = \mathsf{c}p\,(1-p)^{\mathsf{c}-1}$ and $w_2 = 1 - w_1 - (1-p)^{\mathsf{c}}$. We refer to Appendix F.3 for the derivation of this bound.

6.2 Impossibility for strong anonymity

Using Theorem 4, we can derive the following impossibility theorems. We refer to Appendix E for their proofs.

Theorem 5. *For user distribution U_P, with $\hat{\ell} < \mathsf{N}$ and $B < (\mathsf{N}-1) - \epsilon(\eta)$, no protocol $\Pi \in M$ can achieve strong anonymity if $p\hat{\ell} < 1 - \epsilon(\eta)$. Moreover, strong anonymity can not be achieved if $\hat{\ell} \in O(1)$.*

Theorem 5 in words. We confirm that a protocol even with user coordination generally can only provide strong anonymity if it (1) uses a massive amount of bandwidth overhead $B \geq \mathsf{N} - 1$ (intuitively: "for every real message, every other user sends a share"); or (2) satisfies the bound for U_P without compromised nodes from Das et al. [12, 12]. Thus, we confirm that for $B < \mathsf{N} - 1$ their basic trilemma condition (without compromised nodes) holds even against protocols with user coordination. In other words, while user coordination with $B < \mathsf{N}$ strengthens a protocol against compromised parties, it does not suffice for overcoming the basic trilemma condition.

Theorem 6. *For user distribution U_P, Given $p < 1 - \epsilon(\eta)$, $B < (\mathsf{N} - 1) - \epsilon(\eta)$ $\frac{\mathsf{c}}{\mathsf{K}} = const$, no protocol $\Pi \in M$ can achieve strong anonymity if $c > \hat{\ell}^2$ and $\hat{\ell}^2 \in O(log(\eta))$, where $\epsilon(\eta) = 1/\eta^x$ for a positive constant x.*

Theorem 6 in words. If a constant fraction $\frac{\mathsf{c}}{\mathsf{K}}$ of protocol parties is compromised and the protocol does not use a massive bandwidth overhead (see above), then the latency has to grow significantly with the security parameter ($\hat{\ell}$ must grow superlogarithmic in η).

Theorem 7. *For user distribution U_P, given $B < (\mathsf{N} - 1) - \epsilon(\eta)$, no protocol $\Pi \in M$ can achieve strong anonymity if $p \times \max\left\{\hat{\ell} - \mathsf{c}, \frac{\hat{\ell}}{2}\right\} < 1 - \epsilon(\eta)$.*

Theorem 7 in words. For U_P, if the protocol does not use a massive bandwidth overhead (see above), then compromised parties reduce the effective latency in terms of the basic trilemma by a factor of up to two; the more parties can be compromised, the harder it becomes for the protocol.

Theorem 8 (Anytrust Impossibility Theorem). *For user distribution U_P with $\mathsf{K}, \mathsf{N} \in poly(\eta)$, $\mathsf{K} - \mathsf{c} = \gamma \in O(1)$, $\mathsf{K} > \mathsf{c}$, no protocol $\Pi \in M$ can achieve strong anonymity if $\hat{\ell} \leq \mathsf{N} - 1$ or $B \leq \mathsf{N} - 2$ or $\hat{\ell}^2 \leq \mathsf{K} - \gamma$.*

This theorem is similar to Theorem 3 from Section 5. If there are only constant number of honest nodes, strong anonymity is impossible without a large latency overhead or a huge bandwidth overhead or if the latency overhead $\hat{\ell}$ allows each packet to traverse at least $\sqrt{\mathsf{c}}$ parties. The proof is also similar to that of Theorem 3, therefore we skip the proof here.

7 Discussion of results

7.1 Impossibility results

From our impossibility theorems in Sections 5 and 6, we observe that strong anonymity requires a combination of latency overhead and bandwidth overhead – which is very similar to the observations from Das et al. [12]. The strong assumption of user coordination (U.C.) appears to reduce the cost to achieve anonymity, but in most cases does not suffice. As an exception, strong anonymity can always be achieved with user coordination for $B \geq \mathsf{N}$ – even in cases where it is provably impossible for protocols that do not use user coordination.

In Table 1 we compare the impossibility results for protocols with user coordination with those for protocols without user coordination. We compare different cases for the number of compromised nodes c in relation to the latency overhear ℓ and the bandwidth overhead B. Each entry shows under which condition we can prove that strong anonymity is impossible. Recall that the impossibility bounds cannot be tight, as we solely consider the possible paths adversary $\mathcal{A}_{paths}$. Tight bounds would have to also make requirements about the message distributions. Our results are nevertheless comparable to those from [12], since we use the same adversary $\mathcal{A}_{paths}$.

Unified impossibility bound for both user distributions. When comparing our impossibility results for both user distributions, we can represent them with a single unified impossibility condition $\hat{\ell}(p' + \beta) < 1 - \epsilon(\eta)$, where β is the number of noise messages per user per round. For the unsynchronized user message distribution, $\beta = p'B = p - p'$. For the synchronized user distribution, $\beta = \frac{B}{\mathsf{N}} = p'B$, since p' by definition is $\frac{1}{\mathsf{N}}$.

Limitations of our results. In our derivations we do not consider a probabilistic adversary which indeed has a higher chance of deanonymizing users. Additionally, we do not count the cost of user coordination in our results. These factors make our results untight, still giving us a strict lower bound on the cost of anonymity in terms of latency and bandwidth overhead.

We also assume that no user ever goes offline, which means that any restrictions we prove in our protocol model directly translate to both protocols that have an *always online* representation of users and protocols that are more vulnerable. In other words: strong anonymity might be even harder to achieve in practice. This makes our analysis slightly more untight for protocols that don't provide solutions for coping with offline users and set intersection attacks.

Conversely, notions weaker than "strong anonymity", e.g., a partial but robust anonymity set, can be easier to achieve. However, if the cardinality of such a partial set is known in advance our analysis can be easily adapted by reducing the "set of all users" to the partial set and then following our methodology to compute bounds: If N_W is the cardinality of an anonymity set W, our bounds will hold for parameter N_W instead of N.

Table 1. Impossibility Conditions for Anonymous Communication, with number of protocol-nodes K, number of compromised protocol parties c, number of clients N, latency overhead ℓ. In all cases we assume that $\ell < \mathsf{N}$, $1 \leq B < (\mathsf{N}-1) - \epsilon(\eta)$, and $\epsilon(\eta) = 1/\eta^d$ for a positive constant d. We compare different cases for the number of compromised nodes c in relation to the latency overhead ℓ and the bandwidth overhead B. Each entry shows under which condition we can prove that strong anonymity is impossible. Note that we allow protocols with U.C. to utilize a latency of $\hat{\ell} = \ell + 1$ (c.f., Footnote 2).

Cases	U_B without U.C.	U_B, with U.C.	U_P without U.C.	U_P, with U.C.
$\mathsf{c} \geq 0$	$\ell(B+1) < \mathsf{N} - \epsilon(\eta)$	$\hat{\ell}(B+1) < \mathsf{N} - \epsilon(\eta)$	$\ell p < 1 - \epsilon(\eta)$	$\hat{\ell} p < 1 - \epsilon(\eta)$
$0 < \mathsf{c} \leq \ell$	$(\ell - \mathsf{c})(B+1) < \mathsf{N} - \epsilon(\eta)$	$(\hat{\ell} - \mathsf{c})(B+1) < \mathsf{N} - \epsilon(\eta)$	$(\ell - \mathsf{c})p < 1 - \epsilon(\eta)$	$p(\hat{\ell} - \mathsf{c}) < 1 - \epsilon(\eta)$
$\ell < \mathsf{c} \leq (B+1)\ell$	$\ell \in O(1)$	$\hat{\ell}(B+1) < \mathsf{N} - \epsilon(\eta)$	$\ell \in O(1)$	$\hat{\ell} \in O(1)$
$(B+1)\ell < \mathsf{c}$	$\ell \in O(1)$	$\hat{\ell} \in O(1)$	$\ell \in O(1)$	$\hat{\ell} \in O(1)$
$\mathsf{K}/\mathsf{c} \in O(1)$	$\ell \in \log(\eta)$	$\hat{\ell} \in \sqrt{\log(\eta)}$	$\ell \in \log(\eta)$	$\hat{\ell} \in \sqrt{\log(\eta)}$

Table 2. Interesting cases for AC, with number of protocol-nodes K, number of compromised protocol parties c, number of clients N, latency overhead ℓ. The table assumes for all rows $\mathsf{N} \in \Theta(\eta^2)$, $\mathsf{K} \in \Theta(\eta)$, $\ell < \mathsf{K} < \mathsf{N}$ and $B \leq (\mathsf{N}-2)$. Here, ✗ denotes that strong anonymity is provably impossible and (✓) denotes that we could not show this impossibility, i.e., strong anonymity could be possible. In some cases the impossibility proofs rely on additional requirements, i.e., we can only show ✗ if these requirements are met. Note that we allow protocols with U.C. to utilize a latency of $\hat{\ell} = \ell + 1$ (c.f., Footnote 2).

	U_B without U.C.		U_B with U.C.		U_P without U.C.		U_P with U.C.	
Interesting cases	Ano.	Add. req.	Ano.	Add. req.	Ano.	Add. req.	Ano.	Add. req.
$\beta\ell = 1, \ell < \mathsf{K}, \mathsf{c} \in \Theta(\log(\mathsf{K}))$	✗		✗		✗	$p' < \frac{1}{\ell}$	✗	$p' < \frac{1}{\ell}$
$\beta = \frac{1}{\ell}, \ell \in O(1), \mathsf{K} - \mathsf{c} \in \Theta(\eta)$	✗		✗		✗		✗	
$\beta = \frac{1}{\ell}, \ell < \mathsf{c} < \ell^2, \mathsf{K} - \mathsf{c} \in \Theta(1)$	✗		✗		✗		(✓)	
$\beta = \frac{1}{\ell}, \ell^2 \leq \mathsf{c}, \mathsf{K} - \mathsf{c} \in \Theta(1)$	✗		✗		✗		✗	
$\beta = \frac{1}{\sqrt{\ell}}, \ell^2 \leq \mathsf{c}, \mathsf{K} - \mathsf{c} \in \Theta(1)$	✗		✗		✗		✗	
$\beta\ell \in O(1), \ell \leq \log(\mathsf{K}), \mathsf{c} = \mathsf{K}/2$	✗		✗		✗	$p' < \frac{1}{2}$	✗	$p' < \frac{1}{2}$
$\beta > \frac{1}{\log(\eta)}, \ell \geq \log(\mathsf{K}), \mathsf{c} = \mathsf{K}/4$	✗		(✓)		✗		(✓)	
$\ell < \frac{\log(\eta)}{2}, \mathsf{c} = \mathsf{K} - 1$	✗		✗		✗		✗	

7.2 Interesting cases & corner cases

This section discusses some boundary cases and some interesting cases to breathe life into our necessary constraints. We discuss combinations of bandwidth overhead B, latency overhead ℓ, and number c of compromised nodes with respect to the impact of utilizing user coordination (U.C.) in an ACN. In Table 1 we compare the impossibility results for those cases for protocols with user coordination with those cases. Here, ✗ denotes that strong anonymity is provably impossible and (✓) denotes that we could not show this impossibility, i.e., strong anonymity could be possible. In some cases the impossibility proofs rely on additional requirements, i.e., we can only show ✗ if these requirements are met.

Our results are dominated by the universal necessary constraints without any compromisation, i.e., $\hat{\ell}(p' + \beta) < 1 - \epsilon(\eta)$. Hence, the focus of Table 2 is to show which combinations of parameters along the lines $\hat{\ell}(p' + \beta) = 1$ are impossible for which scenario. We illustrate that, while U.C. might lead to strongly anonymous ACNs in some cases, there are interesting cases along the lines of the universal necessary constraints where even ACNs with U.C. cannot achieve strong anonymity.

When we compare the results for protocols without user coordination vs. protocols with user coordination, we compare $\ell = x$ vs. $\hat{\ell} = x$ to induce fairness[2]. For deciding the verdicts, we directly use the lower bounds on δ from our results.

Constant latency, full bandwidth overhead. Let us consider ACNs that send for every real message N shares ($B = N$), which we call full bandwidth overhead. In this case, from our lower bounds on δ we can observe that U.C. has an impact, as no internal node is needed to achieve strong anonymity, as is done in DC-nets [8]. As a consequence, even if there are internal parties but all internal parties are compromised U.C. leaves the possibility of achieving strong anonymity (e.g., along the lines of DC-nets). Without U.C., strong anonymity is impossible if the latency is short ($\ell \in O(1)$). However, when a protocol does not have full bandwidth overhead,

[2] When we allow latency to be $\ell + 1$ for protocols with user coordination to approximate noise generated by internal parties with user noise, we also allow protocols with only user noise to have latency $\ell + 1$. It is unfair to compare them with protocols without user coordination with latency ℓ. Moreover, when $\ell = 0$, there is no intermediate party, so there is no internal noise.

U.C. can not provide strong anonymity without the help of latency overhead and honest intermediate parties.

Almost very high latency, high bandwidth overhead. For high latency bounds $\ell \leq \mathsf{K} - 1$ that are just shy of visiting every node in the ACN ($\ell = \mathsf{K}$), strong anonymity is impossible for synchronized users, even if a high amount of bandwidth overhead $B = \mathsf{N}/\ell$ or $\beta = 1/\ell$ is tolerated. (In Table 2 we use β to unify the impossibility bounds for synchronized and unsynchronized user message distribution, where $\beta = p'B$; for synchronized users, always $p' = 1/\mathsf{N}$.) In Appendix F.1 we provide additional calculations relevant for these corner cases. For the unsynchronized user distribution, strong anonymity is impossible if the rate p' at which real messages are sent per round is low, roughly $p' < 1/\ell$.

Moderate latency, minimal bandwidth overhead. Next, we consider interesting cases where we fix the latency ℓ and consider a bandwidth overhead in such a way that β is along the lines of $\beta\ell = 1$. For the synchronized user distribution, if the latency $\ell \approx \sqrt{\eta} \approx \sqrt{\mathsf{K}}$ and $B = \mathsf{N}/\ell$, our results leave the possibility for strong anonymity only if the total number of compromised parties is less than ℓ, i.e., $\ell > \mathsf{c}$. For the unsynchronized user distribution, for similar latency ($\ell \approx \sqrt{\mathsf{K}}$) and compromisation up to $\mathsf{c} \leq \ell^2$, strong anonymity is possible and the bandwidth overhead can be as low as $B = \beta/p' = O(1)$ for a high rate of real messages (p' is a constant fraction). If all nodes but one are compromised ($\mathsf{c} = \mathsf{K} - 1$), strong anonymity is impossible for both user distributions when $\ell < \sqrt{\mathsf{K}}$, independent of the bandwidth overhead — which confirms our Anytrust impossibility theorem (Theorems 3 and 8).

Log latency, with nearly full bandwidth overhead. Along the line $\beta\ell = 1$, another interesting case is $\ell = \log(\mathsf{K})/2$. In this case, the latency overhead is so low that there is no chance to evade a pervasive adversary that compromises a lot of nodes ($\mathsf{c} \geq \mathsf{K}/2$). In a more specific case, strong anonymity is impossible in a strong compromisation scenario where all nodes but one are compromised ($\mathsf{c} = \mathsf{K} - 1$), regardless of the bandwidth overhead, i.e., for any $\beta < 1 - \epsilon(\eta)$ and $B \leq (\mathsf{N} - 2)$. For a slightly higher latency $\ell \geq 2\log(\mathsf{K})$ and a weak adversary with $\mathsf{c} \leq \mathsf{K}/4$, we cannot exclude the possibility for strong anonymity as long as the universal necessary constraints are satisfied ($\hat{\ell}(p' + \beta) \geq 1$).

In Appendix A, we discuss AC protocols from the literature that utilize some form of user coordination technique, how close they are to our bounds, and the overhead of the user-coordination subprotocols.

8 Conclusion and future work

In this work, we have shown that the anonymity trilemma by Das et al. leaves out AC protocols that utilize what we call user coordination, an ability by which users work together to introduce uncertainty. We extended their analysis and have covered all known ACNs, including DC-nets (and Herbivore and Dissent-AT).

We have shown that even this additional power does not fundamentally change the requirements on latency overhead ℓ and bandwidth overhead B – except that excessive bandwidth on its own is sufficient to provide strong anonymity, independent of latency or even the level of compromisation. In the absence of this extreme case, a combination of latency and bandwidth overhead similar to the results of Das et al. is still necessary. In addition to confirming this crucial insight, our formal analysis yields additional requirements for strong anonymity based on the number of compromised nodes c: if $\mathsf{c} > 0$ then the latency overhead must grow ($\ell \notin O(1)$); if $\mathsf{c}/\mathsf{K} \geq 1/2$ and $\ell \leq \log(\eta)$, then more and more messages must be in the system, i.e., ℓp cannot be constrained by any constant; if $\mathsf{K} - \mathsf{c} \in O(1)$, such as in the Anytrust assumption, then either $\ell > \eta^2$ or $\ell p > \sqrt[4]{\eta}$ are required.

Future work on ACNs can directly build on our insights; our formulas indicate that user coordination in the style of DC-nets (or Herbivore or Dissent-AT) can reduce the gap to the universal necessary constraint ($\ell p \geq 1 - \epsilon(\eta)$) significantly. For closing the gap, however, our results point to ACN designs outside of our wide class of ACNs (see Appendix B.3 for a thorough discussion). Protocol designers thus face a choice: settle for a weaker notion of anonymity, come up with novel techniques, sacrifice reliability or adhere to the latency and bandwidth overheads described in this work.

Acknowledgments

We thank our shepherd Sebastian Angel and anonymous reviewers for their valuable comments. This work has been partially supported by Zurich Information Security Center (ZISC), the European Commission through H2020-DS-2014-653497 PANORAMIX, EPSRC Grant EP/M013-286/1, the National Science Foundation under grant CNS-1719196, CNS-1846316, and the Purdue Research Foundation (PRF) Research Grants.

References

[1] N. Alexopoulos, A. Kiayias, R. Talviste, and T. Zacharias, *MCMix: Anonymous Messaging via Secure Multiparty Computation*, in Proceedings of the 26th USENIX Security Symposium, USENIX Association, 2017, pp. 1217–1234.

[2] M. Ando, A. Lysyanskaya, and E. Upfal, *Practical and Provably Secure Onion Routing*, in Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP), Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 144:1–144:14.

[3] ———, *On the Complexity of Anonymous Communication Through Public Networks*, CoRR arXiv, abs/1902.06306 (2019).

[4] S. Angel and S. Setty, *Unobservable Communication over Fully Untrusted Infrastructure*, in Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI), USENIX Association, 2016, pp. 551–569.

[5] G. R. Blakley and C. Meadows, *Security of ramp schemes*, in Advances in Cryptology, 1985, pp. 242–268.

[6] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, *(leveled) fully homomorphic encryption without bootstrapping*, in Proceedings of the 3rd Innovations in Theoretical Computer Science (ITCS) Conference, 2012, pp. 309–325.

[7] D. Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, Communications of the ACM, 4 (1981), pp. 84–88.

[8] D. Chaum, *The dining cryptographers problem: Unconditional sender and recipient untraceability*, Journal of Cryptology, 1 (1988), pp. 65–75.

[9] C. Chen, D. E. Asoni, D. Barrera, G. Danezis, and A. Perrig, *HORNET: High-speed onion routing at the network layer*, in Proc. ACM Conference on Computer and Communications Security (CCS), 2015, pp. 1441–1454.

[10] H. Corrigan-Gibbs, D. Boneh, and D. Mazières, *Riposte: An anonymous messaging system handling millions of users*, in Proc. 36th IEEE Symposium on Security and Privacy (S&P 2015), 2015, pp. 321–338.

[11] G. Danezis, C. Diaz, C. Troncoso, and B. Laurie, *Drac: An architecture for anonymous low-volume communications*, in Proc. 10th Privacy Enhancing Technologies Symposium (PETS 2010), 2010.

[12] D. Das, S. Meiser, E. Mohammadi, and A. Kate, *Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency - choose two*, in 2018 IEEE Symposium on Security and Privacy (SP), May 2018, pp. 108–126. extended version under https://eprint.iacr.org/2017/954.

[13] R. Dingledine, N. Mathewson, and P. Syverson, *Tor: The Second-Generation Onion Router*, in Proc. 13th USENIX Security Symposium (USENIX), 2004, pp. 303–320.

[14] ———, *Tor: The second-generation onion router*, in Proc. 13th USENIX Security Symposium, 2004.

[15] N. Gelernter and A. Herzberg, *On the limits of provable anonymity*, in Proc. Workshop on Privacy in the Electronic Society (WPES 2013), 2013, pp. 225–236.

[16] S. Goel, M. Robson, M. Polte, and E. Sirer, *Herbivore: A scalable and efficient protocol for anonymous communication*, (2003). https://www.cs.cornell.edu/people/egs/herbivore/herbivore.pdf.

[17] P. Golle and A. Juels, *Dining cryptographers revisited*, in Proc. of Eurocrypt 2004, 2004.

[18] A. Hevia and D. Micciancio, *An indistinguishability-based characterization of anonymous channels*, in Proc. Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008), N. Borisov and I. Goldberg, eds., 2008, pp. 24–43.

[19] K. Jensen, *Colored Petri Nets. Basic Concepts, Analysis Methods and Practical Use.*, vol. 3, 1997.

[20] D. Kesdogan, J. Egner, and R. Büschkes, *Stop-and-go MIXes: Providing probabilistic anonymity in an open system*, in Proc. Information Hiding Workshop (IH 1998), 1998.

[21] L. M. Kristensen, S. Christensen, and K. Jensen, *The practitioner's guide to coloured petri nets*, International Journal on Software Tools for Technology Transfer (STTT), 2 (1998), pp. 98–132.

[22] A. Kwon, D. Lazar, S. Devadas, and B. Ford, *Riffle: An efficient communication system with strong anonymity*, Proceedings on Privacy Enhancing Technologies, 2016 (2016), pp. 115–134.

[23] D. Lazar and N. Zeldovich, *Alpenhorn: Bootstrapping secure communication without leaking metadata*, (2016).

[24] S. Le Blond, D. Choffnes, W. Caldwell, P. Druschel, and N. Merritt, *Herd: A Scalable, Traffic Analysis Resistant Anonymity Network for VoIP Systems*, in Proc. ACM Conference on Special Interest Group on Data Communication (SIGCOMM 2015), 2015, pp. 639–652.

[25] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi, *AnoA: A Framework For Analyzing Anonymous Communication Protocols*, Journal of Privacy and Confidentiality (JPC), 7(2) (2016).

[26] P. Mittal, M. Wright, and N. Borisov, *Pisces: Anonymous communication using social networks*, in Proc. 20th Network and Distributed System Security Symposium (NDSS 2013), 2013.

[27] S. Oya, C. Troncoso, and F. Pérez-González, *Do dummies pay off? limits of dummy traffic protection in anonymous communications*, in Proc. 14th Privacy Enhancing Technologies Symposium (PETS 2014), 2014.

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

[29] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, *Anonymous Connections and Onion Routing*, IEEE J-SAC, 16 (1998), pp. 482–494.

[30] W. Reisig, *Primer in Petri Net Design*, 1st ed., 1992.

[31] T. Ruffing, P. Moreno-Sanchez, and A. Kate, *P2P Mixing and Unlinkable Bitcoin Transactions*, in Proc. 25th Annual Network & Distributed System Security Symposium (NDSS), 2017.

[32] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich, *Vuvuzela: Scalable private messaging resistant to traffic analysis*, in Proc. 25th ACM Symposium on Operating Systems Principles (SOSP 2015), 2015.

[33] D. I. Wolinsky, H. Corrigan-Gibbs, B. Ford, and A. Johnson, *Dissent in Numbers: Making Strong Anony-*

mity Scale, in 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12), 2012, pp. 179–182.

A Implications and scope

Our novel necessary constraints for the core of ACNs with user coordination describe a large set of lower bounds for combinations of bandwidth overhead, latency overhead, resistance to compromised parties, and the degree of anonymity. The rich literature on ACNs contains a few proposals that come close to these novel necessary constraints. This section discusses some of these ACNs, in particular, their usage of user coordination to achieve stronger anonymity and the user coordination online overhead against passive adversaries, i.e., without the overhead of DoS countermeasures. We do not discuss protocols that do not utilize user coordination [1, 22, 24, 28, 32].

Chaum started a line of work on so-called DC-nets [8, 16, 17, 33] that implements anonymous broadcast channels assuming users have agreed on some cryptographic keys with each other or with the protocols parties and have decided on a schedule to ensure that only one real message is sent in each round. As we model the single-recipient setting and assume a passive adversary, for communication overhead analyses we assume variants where broadcast implementations are replaced by directed messages from protocol parties to the dedicated recipient. In particular, for [8, 17], in every round, we assume that each party sends either a real message or a noise message (a share in our model) to the recipient. As a protocol in our model, each client would in each round send packets with the same tag and one of these packets would contain the real message, leading to a bandwidth overhead of N. With $B = N$ and $\ell = 0$, DC-nets satisfy our novel necessary constraints for ACNs with user coordination from Theorem 6, thereby showing tightness of our bounds for this border case.[3] Concerning the complexity of the user coordinations, Chaum proposed [8] a solution where two messages sent at the same time over the broadcast channel would collide. To avoid collisions, he proposed to divide the broadcast into N^2 blocks and to constantly maintain a separate reservation array of size N^2 in the broadcast that is sent in each round. Even if only 1-bit messages are sent, this protocol results in an additional bandwidth overhead of $2N^2$ and one additional round. This bandwidth overhead can also be spread over the latency by spreading the reservation array of the blocks over several rounds.

Herbivore [16] partitions the set of clients into several subsets of size N/q (for some integer $q > 1$) and solely implements DC-nets within a partition, effectively reducing the bandwidth B to the size N/q of a partition. With $B = N/q$ and $\ell = 1$, our results prove that Herbivore cannot achieve the employed AnoA-styled notion of strong sender anonymity, which is easy to see: if the two challenge senders u_0, u_1 are from different partitions, an adversary can easily win. Herbivore also uses the concept of reservations for avoiding collisions, yet also provides several bandwidth-latency sweet spots.

Dissent-AT [33] also reduces DC-nets communication overhead. It relies on K computation servers (the K protocol parties in our model). Assuming that every client has a shared secret with each of the K servers, each client only has to send her real message or share to one of the K servers. Afterwards, in our model, these K servers send their combined shares to the dedicated recipient. Hence, the bandwidth overhead is N messages for each real message, except that these N messages are not sent to N parties as in DC-nets (leading to a communication overhead of N^2) but only to one of the K servers (leading to a communication overhead of N). As we assume a single recipient, in our comparison the bandwidth overhead is $B = \mathsf{N}$ just as for DC-nets. Hence, Dissent-AT satisfies our necessary constraints for ACNs with user coordination from Theorem 6; so, our results do not exclude strong anonymity for Dissent-AT. DISSENT-AT uses a verifiable shuffle among the K servers and results in a periodic latency overhead of K.

Dicemix [31] is outside the scope of our model (see Appendix B.3), as it can mix shares with different tags, yet it nevertheless obeys our bounds. Dicemix aims at removing the scheduling requirements of other DC-nets. Dicemix assumes that each party sends a message, and in our synchronized user distribution, it has to wait for N rounds until real messages arrive. The protocol re-

[3] For DC-nets, [12] did not show necessary constraints under active attacks.

quires 4 communication rounds[4] leading to a latency of $N+4$ in our model, which includes the user coordination's collision-avoidance subprotocol. Every party sends N packets whenever all messages have been collected (in every Nth round); so, the bandwidth overhead (per client) in our model is $B = N$. If we average the overhead ($B' = B/N = 1$) over N rounds, however, Dicemix is close to our universal bound $N = B'\ell \geq N$, hence our results do not rule out strong anonymity, even if almost all other parties are compromised.

All of the above protocols only deal with a boundary condition from our results and their bandwidth overheads are tremendous. None of the known ACNs with user coordination utilize the combination of multi-hop layered encryption feature (as used in mix-nets) with user coordination features that render the real sender's packet indistinguishable from a noise message, even for the recipients. Indeed, there is significant scope for improvement here specially if we need to reduce the bandwidth overhead by introducing some latency overhead.

ACNs with global static synchronization (i.e., U_B with U.C.) effectively introduce large overhead (e.g., N rounds for DC-nets), since each user has to wait for its turn to send a message. Hence, such ACNs are difficult to use with low-latency applications. Moreover, current designs with user coordination (e.g., DC-nets or Dissent-AT) can only then provide a full anonymity[5] set (encompassing all clients) in the Anytrust setting ($\mathsf{c} = \mathsf{K} - 1$) if almost all clients send a dummy message (i.e., $B = \mathsf{N} - 1$). For dynamic user coordination (U_P with U.C.), our results, however, do not exclude strong anonymity for $B < \mathsf{N}$ if sufficient latency is added, as in the third row of Table 2: $B = \beta/p' = 1/(\ell p') = q/\sqrt{\mathsf{K}}$ (for $p' = 1/q$ and a constant q), $\ell = \sqrt{\mathsf{K}}, \mathsf{c} = \mathsf{K}-1, \mathsf{K} = \eta, \mathsf{N} = \mathsf{K}^2$. Such overhead combinations might be interesting for future exploration of ACN designs.

One possible direction is to reconsider the recent mix-net protocol designs [1, 24, 28, 32] in light of user coordination. In particular, our lower bounds indicate that these designs could benefit from incorporating user coordination techniques, which could increase their resistance against compromisation (by increasing the bandwidth overhead B) while reducing latency overhead ℓ. Another possibility, for employing the user coordination, is to consider Riposte design [10], which uses the private information storage primitive. In Riposte, to enable the recipient to point to the exact incoming packet a sender input needs to include a number of elements proportional to the square root of the size of the whole stored database. Using user coordination can allow the Riposte-like design to reduce this bandwidth overhead by sending a smaller number of elements.

B Protocol model for AC protocols

The main purpose of an AC protocol is to let an AC-user (from the set of users $\mathcal{S}$) send information to a recipient (from the set of recipients $\mathcal{R}$). Typically, an AC protocol utilizes a set of nodes (anonymizing parties P) to improve performance and distribute trust. In this work, we consider a global eavesdropping (i.e., passive) adversary $\mathcal{A}$ that can observe the link between any two parties $\mathcal{S} \cup \mathsf{P}$ (including anonymizing parties and users) and has additionally compromised a set of c anonymizing parties $\mathsf{P_c} \subseteq \mathsf{P}$.

We assume that the AC protocol uses cryptographic means (e.g., encryption or secret sharing) to hide the actual message that a *packet* between two parties $P_1, P_2 \in \mathsf{P} \cup \mathcal{S}$ contains. We abstract the leakage of each such packet as the current round number, the direct sender P_1, the direct recipient P_2, and a random identifier for the packet. This leakage indicates that a packet was sent but doesn't leak any content. Consequently, the adversary only sees the challenge message in plain text when it reaches the recipient.

We stress that we do not require the sets $\mathcal{S}$, $\mathcal{R}$, and P to be mutually disjoint. In some protocols from the literature, these sets actually intersect [8, 17, 31]. As we concentrate on sender anonymity, for simplicity we require the set $\mathcal{R}$ of recipients, to be disjoint from $\mathcal{S} \cup \mathsf{P}$.

Next, using a Petri net model, we formally define a generic AC protocol that captures a large class of AC protocols. This section presents an extension of the protocol model from [12] with User coordination. Hence, large parts of the protocol model coincide with the protocol model from [12].

[4] While Dicemix includes integrity protection and self-healing mechanism that leads to $4 + 2f$ communication rounds for one message if f peers are deviating from the protocol, these mechanisms do not kick in if all peers follow the protocol (as even the compromised parties do in our analysis), leading to only 4 communication rounds.

[5] Satisfying strong anonymity implies achieving a full anonymity set.

B.1 Protocol model

This section defines a generic timed colored Petri net [19, 21, 30] M that can be instantiated with a large class of (abstractions of) AC protocols from the literature. We use K as set of parties, $\mathcal{S}$ as the set of users, $P_1, \ldots, P_{\mathsf{K}}$ as the anonymizing (protocol) parties, \$1 as the randomness, R as the recipient of messages, m as a message or packet (containing a real user message, a noise, or being a share) sent by a client or a protocol parties, $T_{\mathcal{S}}$ as transitions for inserting messages into the Petri net (i.e., into the AC protocol), and $T_{P_1}, \ldots, T_{P_{\mathsf{K}}}$ as transitions for sending messages from one party to another. We stress that for every AC protocol, we use the same Petri net M, i.e., the same places, tokens, and transitions. The guards within the transitions can, however, differ; hence, instantiating this generic Petri net M for (the abstraction of) a concrete AC protocol amounts to specifying the guards within the transitions, e.g., by specifying to which party messages are sent next or how much a message should be delayed. As this specification of the generic Petri net M shows, all protocols that can be instantiated by M, in particular these guards, are oblivious to the challenge message or the challenge users.

Next, we introduce the abstraction of packets in our Petri net model. Formally, packets are colored tokens with eight components. Four public components that an adversary can observe are a unique identifier $\mathsf{ID_t}$, the sender prev and the receiver next of a packet, and the time ts at which the packet is activated. The four private components that an adversary cannot observe are the message content msg, some internal protocol meta-data meta, the message's time-to-live $\mathsf{t_r}$, and the share-group tag to which this message belongs (see below).[6] We treat the following list as a definition that we quote verbatim (in light green) from [12] with minor modifications:

Definition 3 (Colored token (green part from [12])). *A colored token is represented by the tuple* $m = \langle \mathsf{msg}, \mathsf{tag}, \mathsf{meta}, \mathsf{t_r}, \mathsf{ID_t}, \mathsf{prev}, \mathsf{next}, \mathsf{ts} \rangle$,
where,

- msg *is the content of the message,*
- meta *is the internal protocol meta-data for this message,*
- $\mathsf{t_r}$ *is the time the message can remain in the network,*
- $\mathsf{ID_t}$ *is a new unique ID generated by each transition for each token by honest parties; dishonest parties instead keep* $\mathsf{ID_t}$ *untouched to allow the adversary to link incoming and outgoing messages,*
- prev *is the party/user that sent the token and* next *is the user/party that receives the token.*
- *Finally,* ts *is the time remaining for the token to be eligible for a firing event (a feature of timed Petri nets). 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.*
- *For allowing user coordination, we introduce an additional field* tag *that allows a token to be* tagged *and several such tokens to contribute to sending one single message. When user coordination is used by the protocol,* msg *field of all the tokens contributing for a single message are populated with* $\perp$*, and the* tag *field of all those tokens are populated with a same tag. We discuss below, how the recipient can retrieve the original message content once he receives a sufficient number of such tokens.*

$\mathsf{ID_t}, \mathsf{prev}, \mathsf{next}, \mathsf{ts}$ are public fields – which means they are always visible to the adversary. However, the fields meta and $\mathsf{t_r}$ are never visible to the adversary. The fields msg and tag can not be observed by the adversary until a packet reaches the recipient.

In case user coordination is used, the field msg does not help to retrieve the message content (because $\mathsf{msg} = \perp$). In this case we use a more complex reconstruction: the recipient has access to a dictionary D (outside the petri-net); when a message reaches the recipient, the recipient queries the dictionary D to retrieve the content of the message. The dictionary has four fields $\langle \mathsf{tag}, \mathsf{msg}, \mathtt{count}, \mathtt{countNeeded} \rangle$. The field msg stores the actual content of the message. The fields $\mathsf{tag}, \mathsf{msg}, \mathtt{countNeeded}$ are already populated (during initialization of the system), whereas the value of $\mathtt{count}$ is set to 0 initially. Every time, the recipient queries the dictionary with $\mathsf{D}[\mathsf{tag}]$, the dictionary increments the value of $\mathtt{count}$ by 1; and only when $\mathtt{count}$ reaches the value of $\mathtt{countNeeded}$ it returns msg. We want to specify here that each token in our petri-net model can contain only one tag.

We define a set Tokens that that contains each pair (t, r), where t is a copy of a colored token and r the round number in which the token was observed. 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

[6] As sender anonymity solely considers one recipient, for simplicity we do not list the final recipient of the message in the private part.

token and r the round number in which the token was observed.

Places. We treat the list of places as a definition and quote the definition verbatim from [12].

Any AC protocol with K parties $\mathsf{P} = \{P_1, \ldots, P_{\mathsf{K}}\}$ consists of the following places:

– $\mathcal{S}$: A token in $\mathcal{S}$ denotes a user message (real or noise) which is scheduled to enter the network after ts rounds.

– \$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 $P_i \in \mathsf{P}$: A token in P_i denotes a message which is currently held by the party $P_i \in \mathsf{P}$.

– R: A token in R denotes a message which has already been delivered to a recipient.

```
**T_X on tokens** $q = \langle \mathsf{msg}, \mathsf{tag}, \_, \mathsf{t_r}, \mathsf{ID_t}, \_, \mathsf{prev}, \mathsf{ts}\rangle$
**from** $X \in \mathcal{S} \cup \mathsf{P}$, \$ **from** \$1:
    $(P', \mathsf{meta}') = f_\Pi(q, \$)$ ; $r$ = current round
    **if** $\mathsf{t_r} = 0$ **then** $P' = R$
    **if** $X \in \mathsf{P}$ and $X$ is compromised **then** $\mathsf{ID_t}' = \mathsf{ID_t}$
    **else** $\mathsf{ID_t}'$ = a fresh randomly generated ID
    $t = \langle \mathsf{tag}, \mathsf{meta}', \mathsf{t_r} - 1, \mathsf{ID_t}', P_i, P', 1\rangle$
    **if** $P' \neq R$ **then** obs $= \langle \_, \_, \_, \_, \mathsf{ID_t}', \mathsf{prev}, P', 1\rangle$
    **else** obs $= \langle \mathsf{msg}, \mathsf{tag}, \_, \_, \mathsf{ID_t}', \mathsf{prev}, P', 1\rangle$
    $\mathsf{Tokens} = \mathsf{Tokens} \cup \{(\text{obs}, r)\}$
**Output:** token $t$ at $P'$

$f_\Pi$: a function provided by $\Pi$ to choose $P'$ and to keep
state meta.

**Reconstruct**(tag):
    **if** tag $= \bot$ or D[tag] does not exist **then return** $\bot$
    D[tag].count = D[tag].count + 1
    **if** D[tag].count = count.countNeeded **then return**
    D[tag] **else return** _
```

Fig. 3. Transitions in the Petri net model M

Transitions. At the beginning of the execution, the challenger specifying the set $\mathcal{S}$ on behalf of the AC protocol. The other places are initialized as empty. Transferring a message from one party to another party is formalized by executing a transition that modifies the *configuration* of the Petri net by consuming a token from one place to producing a token in another place. The Petri net M includes the following transitions, for which the Figure 3 presents the pseudocode. Again, we treat the following list as a definition that we quote verbatim from [12] with minor modifications:

– $T_{\mathcal{S}}$: takes a token $\langle \mathsf{msg}, \mathsf{tag}\_, \_, \_, \_, u, \mathsf{ts}\rangle$ from $\mathcal{S}$ and a token from \$1 to write $t = \langle \mathsf{msg}, \mathsf{tag}, \mathsf{meta}, \ell, \mathsf{ID_t}, u, P_i, \mathsf{ts} = 1\rangle$ to P_i; the values of i and meta are decided by the AC protocol.

– T_{P_i}: takes a token $\langle \mathsf{msg}, \mathsf{tag}, \mathsf{meta}, \mathsf{t_r}, \mathsf{ID_t}, \_, P_i, \mathsf{ts}\rangle$ from P_i and a token from \$1 to write $t = \langle \mathsf{msg}, \mathsf{tag}, \mathsf{meta}', \mathsf{t_r} - 1, \mathsf{ID_t}', P_i, P', 1\rangle$ to P'. If P_i is an honest party $\mathsf{ID_t}'$ is freshly generated, but if P_i is a compromised party $\mathsf{ID_t}' = \mathsf{ID_t}$. The place $P' \in \{P_1, \ldots, P_{\mathsf{K}}\} \cup \{R\}$ and meta' are decided by the AC protocol, except when $\mathsf{t_r} = 0$, P' always is R.

The execution of each transition is followed by adding a pair (t', r') to the set Tokens, with t' being a copy of the produced token t without the fields meta and $\mathsf{t_r}$ and r' being the current round number. Moreover, if the place where t was produced is not in $\mathcal{R}$ also the field msg is not contained in t'.

B.2 Expressing protocols

The generic Petri net M captures a large class of AC protocols. This section discusses the expressivity of M in general and for a few particular interesting cases.

M can model mix networks and onion routing protocols. Abstractions of stop-and-go mix [20] that use a discrete distribution and AC protocols with sophisticated path selection algorithms [11, 26] can be directly encoded in M. AC protocols that are not round-based, e.g., Tor [14], can be abstracted as round-based AC protocols, since we solely use M for impossibility results and making the AC round-based only strengthens the anonymity property.

Users as protocol parties. There are peer-to-peer AC protocols, such as dining cryptographers networks (DC-nets [17, 31]), in which users constitute relays. As in the proof of Claim 2, we model users in such AC protocols with two nodes: one user node and one protocol node. In this way, the latency is increased by one, but in many cases that is not important.

Splitting and recombining messages. Our generic Petri net M can abstract AC protocols that split and recombine messages. In Claim 2, we prove that against our adversary the protocol always benefits from letting the user split the message, which we capture. Moreover, our model capture AC protocols that let the recipient recombine messages. Recombination at an earlier internal protocol node does not help against our adversary. Recall that our adversary tries to follow all paths and throws a random coin if the challenge message cannot be uniquely traced back to a user. Recombining the shares earlier reduces the chance of each share mixing with other messages; hence, it cannot increase our adversary's success probability. As a result, we can overapproximate internal splitting and recombination as user splitting and recombination at the recipient.

Broadcasting messages. We prove that internal parties creating messages is less strong than users creating messages against our concrete adversary (Claim 2); hence, we can overapproximate protocols that internally broadcast messages with protocols where the users broadcast messages. We include protocols in which a user can send multiple messages with the same tag.

B.3 Discussion about user coordination assumptions

In Section 3.6, we make three assumptions regarding the protocol model. Most ACNs from the literature are consistent with these assumptions.

The first assumption is that among the shares employed to reconstruct a message at least one must be sent by the message sender. This follows from our assumption that the messages are unavailable while User Coordination gets established; if senders were allowed to know and transmit their messages during setup, the whole protocol could take place during the setup phase.

The second assumption is that no share can take part in reconstructing two separate messages. Although concepts such as Ramp secret-sharing [5] from the cryptographic literature indeed offer the possibility to extract multiple shared messages from a given set of shares, it requires messages to be known in advance. In general, reusing the same share will introduce confidentiality issues similar to a two-times pad. Dicemix [31] uses such a technique where shares of different messages are mixed and is thus outside our model. Nevertheless, Dicemix utilizes n^2 shares for reconstructing n messages; so, it does not break our impossibility bounds. Recent mailbox-based schemes like Riposte [10] are within our protocols model.

The third assumption is indeed interesting. It expects that a compromised party will always be able to map its outgoing packets with its incoming packets. Although this is trivially correct when there is one incoming packet, the assumption focuses on the question when there are two or more incoming packets. It suggests that the party cannot permute these multiple incoming packets such that it itself cannot determine the employed permutation. Performing non-interactive MPC using fully homomorphic encryption (FHE) [6] may enable a node to permute message locally (i.e., without introducing bandwidth and latency overheads) without determining the permutation. This is highly inefficient for current FHE mechanisms as the evaluation circuit depth will be at least logarithmic in the number of users. Nevertheless, it presents an interesting avenue for future ACN design.

B.4 Construction of a concrete adversary

We use the same adversary $\mathcal{A}_{paths}$ as in the work of Das et al. [12]. As we consider sender anonymity, the adversary can start its analysis of all observations from the challenge message that it observes at the recipient. The adversary $\mathcal{A}_{paths}$ constructs all possible paths from which the challenge message could have originated. Recall that in the sender anonymity game the adversary knows two candidate senders u_0 and u_1. So, the adversary checks whether there is a possible path from the challenge message to u_0 and to u_1. If there is no path to one of then, say u_b, the adversary chooses the other challenge sender u_{1-b}. If there is a path to both of them, $\mathcal{A}_{paths}$ makes a random choice.

More precisely, let $(t', r') \in \mathsf{Tokens}$ be an adversary observation, with t' being the colored token that was observed in round r'. Let r be the round at which the challenge message arrives. Fix $j \in \{0, 1\}$, and let a *possible path for* u_j be a path from a challenge user u_j to the recipient R such that the path is at most ℓ elements long. Observe that if the challenge bit is b the there is at least one possible path for u_b; there has to be a path from u_b to the recipient R. We quote the precise definition from [12] with minor modification.

$$\begin{aligned}S_j = \{ & p = (t_1.\mathsf{prev}, \ldots, t_k.\mathsf{prev}, t_k.\mathsf{next}) : \\ & ((t_1, r_1), \ldots, (t_k, r_k)) \in \mathsf{Tokens} \text{ s.t. } k \le \ell \\ & \land t_1.\mathsf{prev} = u_j \land t_k.\mathsf{next} = R \\ & \land (t_k.\mathsf{msg} = \mathtt{Chall} \lor \mathsf{D}[t_k.\mathsf{tag}] = \mathtt{Chall}) \\ & \land \forall_{i \in \{1,\ldots,k-1\}} (t_i.\mathsf{next} = t_{i+1}.\mathsf{prev} \land r_{i+1} = r_i + 1 \\ & \land (\ \exists t'_{i+1} : (t'_{i+1}, r_{i+1}) \in \mathsf{Tokens} \land t'_{i+1}.\mathsf{prev} = t_i.\mathsf{next} \\ & \land t'_{i+1}.\mathsf{ID_t} = t_i.\mathsf{ID_t}) \Rightarrow t'_{i+1} = t_{i+1})\}\end{aligned}$$

Definition 4 (Adversary $\mathcal{A}_{paths}$). *Given a set of users $\mathcal{S}$, a set of protocol parties* P *of size* K*, and a number of possibly compromised nodes* c*, the adversary $\mathcal{A}_{paths}$ proceeds as follows: 1. $\mathcal{A}_{paths}$ selects and compromises* c *different parties from* P *uniformly at random. 2. $\mathcal{A}_{paths}$ chooses two challenge users $u_0, u_1 \in \mathcal{S}$ uniformly at* *random. 3. $\mathcal{A}_{paths}$ 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 $\mathcal{A}_{paths}$ returns $1-i$. Otherwise, it returns 0 or 1 uniformly at random.*

We stress that $\mathcal{A}_{paths}$ does (per run) not take any probabilities into account. Even if in a particular run it is overwhelmingly more likely that u_b sent the message but there is a negligible chance that u_{1-b} sent the message, $\mathcal{A}_{paths}$ does not decide for either of them and randomly picks one. Moreover, if $\mathsf{c} = 0$, $\mathcal{A}_{paths}$ only constitutes a *non-compromising* global network-level adversary, which compromises no protocol parties yet listens on all links between nodes. If $\mathsf{c} > 0$, $\mathcal{A}_{paths}$ is a *partially compromising* global network-level adversary.

B.5 A possible instance of the Oracle functionality

Figure 4 describes a possible instantiation of our oracle O that our ideal protocol uses. The oracle is initialized before the protocol starts; The oracle provides two main functionalities QueryPaths() and QueryForMessage() as defined in Figure 4.

C Impossibility bounds for protocols without user coordination

The work by Das et al. [12] derives impossibility results for mix nets. This protocol class rules out ACNs with user coordination, like DC-nets. This section summarizes their bounds on anonymity for mix nets. Using Invariant 1 and adversary $\mathcal{A}_{paths}$, Das et al. derive the following lower bounds on the adversarial advantage δ against protocols without user coordination for synchronized (U_B) and unsynchronized (U_P) user message distributions.

Theorem 9 ([12]). *For user distribution U_B, even with $\mathsf{c} = 0$, no protocol $\Pi \in M$ can provide δ-sender anonymity, for any $\delta < 1 - f_\beta(\ell)$, where $f_\beta(x) = \min(1, ((x + \beta\mathsf{N}x)/(\mathsf{N}-1)))$.*

Theorem 10 ([12]). *For user distribution U_B, no protocol $\Pi \in M$ can provide δ-sender anonymity, for any*

$$\delta < \begin{cases} 1 - [1 - \binom{\mathsf{c}}{\ell}/\binom{\mathsf{K}}{\ell}]f_\beta(\ell) & \mathsf{c} \geq \ell \\ 1 - [1 - 1/\binom{\mathsf{K}}{\mathsf{c}}]f_\beta(\mathsf{c}) - f_\beta(\ell - \mathsf{c}) & \mathsf{c} < \ell \end{cases}$$

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

```
Paths := set of paths; Delays⟨message, delay⟩;
MessagePaths⟨message, path⟩;
MessageTags⟨message, tag⟩;

Initialize(Parties $P$, users $U$, input messages $I_U$,
protocol definition $\Pi$, latency $\ell$):
  PathsConfigs ← set of all possible path configuration
  (arrangements of parties in $P$)
  DelaysConfigs ← Set of all possible delay (of messages)
  configuration of $I_U$
  $P_{\text{global}} \leftarrow 0$
  for each (PathsConfig, DelaysConfig) in
  (PathsConfigs, DelaysConfigs) do
    PathsMaps ← set of all possible mappings for
    messages to paths for the given DelaysConfig and
    PathsConfig
    TagsMaps ← all possible valid tags for messages
    mapping noise messages to real messages for the
    purpose of user coordination
    for each (PathsMap, TagsMap) ∈
    (PathsMaps, TagsMaps) do
      $P_{\text{local}}$ ← the probability of satisfying Invariant 2
      by protocol $\Pi$
      if $P_{\text{local}} > P_{global}$ then
        $P_{\text{global}} \leftarrow P_{\text{local}}$; Paths ← PathsConfig
        MessagePaths ← PathsMaps
        MessageTags ← TagsMap
        Delays ← DelaysConfig

QueryPaths():
  return Paths

QueryForMessage(message $m$):
  delay ← Delays.Get($m$) ; tag ← MessageTags.$Get$($m$)
  path ← MessagePaths.Get($m$)
  return (path, delay, tag)
```

Fig. 4. Instance of Oracle Functionality

Theorem 11 ([12]). *For user distribution U_P, even with $\mathsf{c} = 0$, no protocol $\Pi \in M$ can provide δ-sender anonymity, for any $\delta < 1 - \left(\frac{1}{2} + f_p(\ell)\right)$, where $f_p(x) = \min(1/2,\ 1 - (1-p)^x)$ for a positive integer x.*

Theorem 12 ([12]). *For user distribution U_P, no protocol $\Pi \in M$ can provide δ-sender anonymity, for any*

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

where $f_p(x) = \min(1/2, 1-(1-p)^x)$ *for a positive integer* x*.*

The impossibility conditions provided by the classical bounds are summarized in Table 3.

Table 3. Impossibility Results for Anonymous Communication (Mix-nets), with the number of protocol-nodes K, number of compromised protocol parties c, number of clients N, and message-threshold T, expected latency ℓ' per node, dummy-message rate β. In all cases we assume that $\ell < \mathsf{N}$ and $\beta\mathsf{N} \geq 1$ and $\epsilon(\eta) = 1/\eta^d$ for a positive constant d.

dist.	Compromisation	Latency&Bandwidth
U_B	$\mathsf{c} = 0$	$2\ell\beta < 1 - \epsilon(\eta)$
U_B	$\mathsf{K} > \ell > \mathsf{c} \in O(1)$	$2(\ell - \mathsf{c})\beta < 1 - \epsilon(\eta)$
U_B	$\mathsf{K} > \ell > \mathsf{c} \in poly(\eta)$	$2\ell\beta < 1 - \epsilon(\eta)$
U_B	$\mathsf{K} > \mathsf{c} \geq \ell$	$2\ell\beta < 1 - \epsilon(\eta)$ or $\ell \in \mathcal{O}(1)$
U_P	$\mathsf{c} = 0$	$2\ell p < 1 - \epsilon(\eta)$
U_P	$\mathsf{K} > \ell > \mathsf{c} \in O(1)$	$2(\ell - \mathsf{c})p < 1 - \epsilon(\eta)$
U_P	$\mathsf{K} > \ell > \mathsf{c} \in poly(\eta)$	$2(\ell - \mathsf{c})p < 1 - \epsilon(\eta)$
U_P	$K > \mathsf{c} \geq \ell$	$2\ell p < 1 - \epsilon(\eta)$ or $\ell \in \mathcal{O}(1)$

D Proofs of claims from Section 4

Proof of Claim 1. To prove the above, we need to prove that anonymity is broken whenever either of (i) or (ii) is false.

Whenever (i) is false, the set T is empty. Thus, the challenge message could not have been sent by the u_{1-b}.

For (ii) of the invariant to be false, both (ii.a) and (ii.b) have to be false. Note here, (ii.a) directly implies anonymity, because if one of the shares of the challenge message is dispatched by u_{1-b} within rounds $\{(r-\ell), \dots, (r-1)\}$ there is no way for the adversary to distinguish between the challenge users.

If (ii.a) is false, (ii.b) can be false in the following ways:

1. no share of the challenge message passes through an honest node: When the adversary backtracks the paths of the shares of the challenge message starting from the recipient, the path will never cross the paths of any message from u_{1-b} at an honest node. So, $\mathcal{A}_{paths}$ can see that none of the messages from u_{1-b} is a share of the challenge message; u_{1-b} could not have sent the challenge message and hence $\mathcal{A}_{paths}$ wins.

2. At least one of the shares of the challenge message sent at $t \in T$ passes through one or more honest nodes at times t', but $\nexists\, t'$ such that $t' \in \{min(T), (r-1)\}$: Following the same reasoning as above, we see that paths after round $min(T)$ can be ambiguous, but there is no message from u_{1-b} before $min(T)$. So, none of them will mix with any of the shares of the challenge message. Thus, $\mathcal{A}_{paths}$ wins.

3. no message from u_{1-b} sent at $t \in T$ passes through an honest node: Similar to previous cases, when the adversary backtracks the paths of the shares of the challenge message starting from the recipient, no path will cross the paths of the messages from u_{1-b} at an honest node. So, no message from u_{1-b} could have been a share of the challenge message.

4. At least one of the messages from u_{1-b} sent at $t \in T$ passes through one or more honest nodes at times t', but $\nexists\, t'$ such that $t' < r$: Following the same reasoning as above cases, we see that paths after round r can be ambiguous, but the challenge message is already delivered at round r. So, none of them will mix with any of the shares of the challenge message.

In all cases where (ii.b) is false, $\mathcal{A}_{paths}$ wins with probability 1, (assuming that (ii.a) is also false). □

Proof of Claim 2. We prove this claim by construction. Given a protocol Π_1 we want to construct a protocol Π_2 that satisfies the invariant with at least the same probability as Π_1. Once, an internal noise message is created, the content of the message can not be modified (although, it can be re-encrypted with different keys or decrypted), the message has to be delivered to the recipient. Additionally an internal noise message can remain in the system for $\min(\ell, z)$. where z is the latency bound for the message tag the message wants to use. Thus, having a user send a message "costs" as much as having internal nodes create the message. (Any internal noise message created not as a share of a user message will not influence the probability of the invariance being true.)

We can consider two different cases for an internal noise message:

1. **A dishonest node creates the noise message:** since, messages can not mix at a dishonest node, this does not help. Instead, a message sent by a user could help the protocol.

2. **An honest node creates the noise message:** This can definitely help the protocol. However, if a user creates the noise one round before and sends it to the given internal node in the current round, that is at least as good as a noise message created by the node in the current round.

Hence, for each internal noise message m (created at round r) in Π_1 , we make Π_2 send a noise message from a user (picked uniformly at random) at round $r-1$. And, because of the reasons explained above, Π_2 will have at least the same probability as Π_1 in satisfying Invariant 2. However, Π_2 now uses latency overhead $\ell{+}1$ for the messages corresponding to the internal noises in Π_2 that uses latency overhead ℓ. □

Proof of Claim 3. We want to prove our claim by contradiction. Suppose, there exists a protocol Π, given a latency ℓ, satisfies Invariant 2 with a higher probability than Π_{ideal} (that uses latency $\ell+1$), against the adversary $\mathcal{A}_{paths}$. By Claim 2, we can construct a protocol Π_{new} where every message is created by some user $u \in \mathcal{S}$, and allow Π_{new} to use a latency of $\hat{\ell} = \ell + 1$; and Π_{new} will have a probability at least as much as Π_{ideal} to satisfy the invariant.

Now we construct a new protocol Π_{hybrid}, which exactly follows the strategy of Π_{ideal} with one exception: for a given message Π_{hybrid} selects the time delay t same as Π_{new}, instead of querying it from oracle O of Π_{ideal}.

The ideal strategy for ensuring that at least one honest party is on at least one the path of the messages from u_{1-b} is to ensure that as many distinct parties as possible are on all the paths combined. Similarly, the possibility of having an honest party of the paths of the shares of the challenge message is also maximized by maximizing the number of distinct parties on all those paths combined.

Similarly, the ideal strategy for obfuscating the challenge sender with user coordination is by maximizing the number of users sending shares for the challenge message. Since the user distribution is the same for both Π_{new} and Π_{hybrid}, Π_{hybrid} is at least as successful in satisfying the invariant due to the oracle.

For both Π_{new} and Π_{hybrid}, the times when messages are sent and the time delays are same, and hence, for every message the path length is same for both Π_{new} and Π_{hybrid}. However, Π_{hybrid} decides the number of paths, and distribution of the protocol parties on those paths by querying the oracle. Hence, Π_{hybrid} has a probability of satisfying Invariant 2 at least as high as Π_{new}.

Now, if we compare Π_{hybrid} and Π_{ideal} : they follow the same strategy. But Π_{ideal} picks the time delay t for any message from oracle O such that t is *optimal*. Hence, Π_{ideal} satisfies Invariant 2 with probability at least as high as Π_{hybrid}. Thus, Π_{new} does not satisfy Invariant 2 with a higher probability than Π_{ideal}. □

E Deferred proofs

Proof of Theorem 2. We know,
$\delta \geq \left(1 - \frac{B}{\mathsf{N}-1}\right)\left[1 - \frac{(\tau+1)\mathsf{N}-B\hat{\ell}-\hat{\ell}}{\mathsf{N}} g(\tau) - \frac{B\ell+\ell-\tau\mathsf{N}}{\mathsf{N}} g(\tau+1)\right]$.

First, we observe that, if $B\hat{\ell}+\hat{\ell} < \mathsf{N} - \frac{1}{\eta^x}$, τ is zero, and hence, $g(\tau)$ is zero. Moreover, $\frac{B\hat{\ell}-\hat{\ell}}{\mathsf{N}} < 1 - \frac{1}{\mathsf{N}\eta^x} =$ not overwhelming. Which means δ cannot be negligible. Now,

$$B\hat{\ell}+\hat{\ell} < \mathsf{N} - \epsilon(\eta) \impliedby (B+1)\hat{\ell} < \mathsf{N} - \epsilon(\eta).$$

We additionally need both $g(\tau)$ and $g(\tau+1)$ to be overwhelming to achieve strong anonymity. When $c \geq (B{+}1)\hat{\ell}$, both $\tau(\hat{\ell}{+}1)$ and $(\tau{+}1)\hat{\ell}$ have to be in $\omega(1)$ (i.e., not in $O(1)$), in order for $g(\tau)$ and $g(\tau+1)$ to become overwhelming. We know that $B < \mathsf{N}-1 \implies \frac{B}{\mathsf{N}} < 1$. If $\hat{\ell}$ is in $O(1)$,
$\tau = \lfloor \frac{B\hat{\ell}+\hat{\ell}}{\mathsf{N}} \rfloor = \lfloor \left(\frac{B}{\mathsf{N}}\hat{\ell} + \frac{\hat{\ell}}{\mathsf{N}}\right) \rfloor \leq (\ell+1) \in O(1)$.
Hence, $\tau\hat{\ell}$ is also in $O(1)$. Therefore, $g(\tau)$ and $g(\tau+1)$ are not overwhelming. □

Proof of Theorem 3. We know, $\tau = \lfloor \frac{B\hat{\ell}+\hat{\ell}}{\mathsf{N}} \rfloor < \hat{\ell}$. When $\hat{\ell}^2 < \mathsf{K} - \gamma$,

$$\frac{\binom{\mathsf{c}}{\tau\hat{\ell}}}{\binom{\mathsf{K}}{\tau\hat{\ell}}} \geq \frac{\binom{\mathsf{c}}{\hat{\ell}^2}}{\binom{\mathsf{K}}{\hat{\ell}^2}} \geq \frac{\mathsf{c}!\,(\mathsf{K}-\hat{\ell}^2)!}{(\mathsf{c}-\hat{\ell}^2)!\,\mathsf{K}!} \geq \frac{(\mathsf{K}-\gamma)!\,\gamma!}{\mathsf{K}!}$$

For $\gamma \in O(1)$, the above quantity is always non-negligible. Hence, $g(\tau)$ is never overwhelming. Therefore, δ cannot be negligible unless $\hat{\ell} \geq \mathsf{N} - negl(\eta)$ or $B \geq \mathsf{N} - 1 - negl(\eta)$. □

Proof of Theorem 4. Suppose u_0 and u_1 are challenge users, and u_b sends the challenge message. The challenge reaches the recipient at round r. The challenge message can have up to $B = \frac{p-p'}{p'}$ additional shares (excluding the share sent by u_b). Ideally, we want u_{1-b} to send at least one of the $\frac{p-p'}{p'}$ shares. If not, we at least want u_{1-b} to send at least one message in $[r-\hat{\ell}, r-1]$, that passes through an honest node before round r.

By Invariant 2, only the shares sent in rounds $\{(r-\hat{\ell}), \ldots, (r-1)\}$ can contribute to anonymity. Therefore, the number of shares for the challenge message is bounded by $B_{\text{eff}} = min(B, \hat{\ell}p\mathsf{N} - 1)$.

The probability that u_{1-b} sends at least one message within a span of $\hat{\ell}$ rounds is upper bounded by $f_p^{SA}(\ell)$ as explained in Appendix F.2. Moreover, u_{1-b} can not send more than $\hat{\ell}$ messages in $\hat{\ell}$ rounds. Thus, we can derive:

$$
\begin{aligned}
&\Pr\left[\text{Invariant 2 is true}\right]\\
&\leq \Pr\left[u_{1-b} \text{ sends a share of the challenge message.}\right]\\
&\quad + Pr[u_{1-b} \text{ sends no shares of the challenge message}\\
&\qquad \wedge u_{1-b} \text{ sends a message in the given span of round } \hat{\ell}]\\
&\quad \times Pr[\text{Some share of the challenge message visits honest}\\
&\qquad \text{node and some message from } u_{1-b} \text{ visits honest node}]\\
&\leq \frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right) \times Pr[u_{1-b} \text{ sends at least}\\
&\qquad \text{one message in } \{(r-\hat{\ell}), \ldots, (r-1)\}]\\
&\quad \times Pr\left[\text{At least one honest node in } \hat{\ell} \text{ paths}\right]\\
&\leq \frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right) \times g(\hat{\ell}) \times f_p^{SA}(\hat{\ell})
\end{aligned}
$$

By Claim 1, whenever Invariant 2 is not satisfied the adversary wins, bounding the adversary's advantage by:

$$
\begin{aligned}
\delta &\geq 1 - \Pr\left[\text{Invariant 2 is true}\right]\\
&\geq \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\left[1 - g(\hat{\ell}) \times f_p^{SA}(\hat{\ell})\right].
\end{aligned}
$$

□

Proof of Theorem 5. If $B < (\mathsf{N}-1) - \epsilon(\eta)$, $\frac{B}{\mathsf{N}-1}$ will be less than $1 - neg(\eta)$. Hence, $H = \left(1 - g(\hat{\ell}) \times f_p^{SA}(\hat{\ell})\right)$ has to be negligible to achieve strong anonymity. However, this is a generic lower bound on δ, and from Appendix F.2 we know that it is sufficient to consider $\left(1 - (1-p)^{\hat{\ell}}\right)$ instead of $f_p^{SA}(\hat{\ell})$. Hence, we require $H' = \left(1 - g(\hat{\ell}) \times \left(1 - (1-p)^{\hat{\ell}}\right)\right)$ to be negligible for the protocol to achieve strong anonymity. When $p\hat{\ell} < 1 - \epsilon(\eta) \implies (1-p)^{\hat{\ell}} < 1 - \epsilon(\eta)$, H' can never be negligible, and consequently, δ can never be negligible.

Even when $(1-p)^{\hat{\ell}}$ is negligible, $g(\hat{\ell})$ has to be overwhelming as well to achieve strong anonymity in case $\mathsf{c} > \ell$, which implies $\left[\binom{\mathsf{c}}{\hat{\ell}^2}\Big/\binom{\mathsf{K}}{\hat{\ell}^2}\right]$ has to be negligible (since $\mathsf{c} \geq \hat{\ell}^2 \implies \mathsf{c} \geq \hat{\ell}^2$), to achieve strong anonymity. $\binom{\mathsf{c}}{\hat{\ell}^2}\big/\binom{\mathsf{K}}{\hat{\ell}^2}$ can never be negligible if $\hat{\ell}^2 \in O(1)$.

When $\mathsf{c} < \hat{\ell}$, We need $\left[1 - 1/\binom{\mathsf{K}}{\mathsf{c}}\right]$ to be overwhelming to achieve strong anonymity. This means, we need the term $1/\binom{\mathsf{K}}{\mathsf{c}}$ to be negligible and that never happens for a constant c. If $\hat{\ell} \in O(1)$, c is also $O(1)$, since $\hat{\ell} > \mathsf{c}$ by our assumption. And, that shows that our theorem holds for $\mathsf{c} < \hat{\ell}$ as well.

Finally consider the case $\hat{\ell} \leq \mathsf{c} < \hat{\ell}^2$. For a constant ℓ, if a constant c can provide adversary better advantage, the adversary will choose to compromise fewer protocol parties even though he can compromise more. Therefore, Whenever we have constant $\hat{\ell}$, it is impossible to achieve strong anonymity, since it is impossible even for c being as small as 1. □

Proof of Theorem 6. If $B < \mathsf{N} - 1 - \epsilon(\eta)$, $\left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)$ cannot be negligible. In that case, both $f_p^{SA}(\hat{\ell})$ and $g(\hat{\ell}) = 1 - \binom{\mathsf{c}}{\hat{\ell}^2}/\binom{\mathsf{K}}{\hat{\ell}^2}$ have to be overwhelming to make $\delta \geq \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\left[1 - g(\hat{\ell}) \times f_p^{SA}(\hat{\ell})\right]$ negligible. For $\mathsf{c} > \hat{\ell}^2$ and $\frac{\mathsf{c}}{\mathsf{K}} = const = \frac{1}{y}$,

$$
\begin{aligned}
\frac{\mathsf{c} - \hat{\ell}^2}{\mathsf{K} - \hat{\ell}^2} > \frac{1}{y} \iff \left(\frac{\mathsf{c} - \hat{\ell}^2}{\mathsf{K} - \hat{\ell}^2}\right)^{\hat{\ell}^2} &> \left(\frac{1}{y}\right)^{\hat{\ell}^2}\\
\implies \frac{\mathsf{c}\ldots(\mathsf{c} - \hat{\ell}^2)}{\mathsf{K}\ldots(\mathsf{K} - \hat{\ell}^2)} &> \left(\frac{1}{y}\right)^{\hat{\ell}^2}
\end{aligned}
$$

$\left(\frac{1}{y}\right)^{\hat{\ell}^2}$ cannot be negligible for $\hat{\ell}^2 \in O(log(\eta))$. □

Proof of Theorem 7. When $B < (\mathsf{N}-1) - \epsilon(\eta)$, $\left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)$ can never be negligible. Additionally, because $p(\hat{\ell} - \mathsf{c}) < 1 - \epsilon(\eta)$, $(1-p)^{\hat{\ell}-\mathsf{c}}$ can not be negligible. Therefore, to achieve strong anonymity, $\left(1 - (1-p)^{\mathsf{c}}\right)$ and $\left[1 - 1/\binom{\mathsf{K}}{\mathsf{c}}\right]$ has to be overwhelming – that is not possible if $p\mathsf{c} < 1 - \epsilon(\eta)$. [Here we use the knowledge from Appendix F.2 to use $(1-p)^{\hat{\ell}-\mathsf{c}}$ instead of $f_p^{SA}(\hat{\ell} - \mathsf{c})$ and $\left(1 - (1-p)^{\mathsf{c}}\right)$ instead of $f_p^{SA}(\mathsf{c})$.]

Finally, note that the adversary can always choose to compromise less than c nodes and thus would choose to compromise $\frac{\hat{\ell}}{2}$ at most to maximize the advantage. □

F Interesting calculations

F.1 A tighter special case for Theorem 1

When $\tau = 0$ and $\mathsf{c} < \hat{\ell}$, we can derive a more precise bound than the one in Theorem 1. Since $\tau = 0$, there is at most one message sent by u_{1-b} in a span of $\hat{\ell}$ rounds. There is a chance that u_{1-b} does not send a message, the invariants are not satisfied (and the adversary wins) in that case. When u_{1-b} sends a message, the invariants are satisfied only if the whole path of the message is not compromised. However, since $\mathsf{c} < \hat{\ell}$, the adversary can not compromise a whole path of length $\hat{\ell}$. Therefore, the adversary has a chance to break the invariants if the message from u_{1-b} is dispatched in $\{r - \mathsf{c}, \ldots, r-1\}$. If the message is sent by u_{1-b} in $\{r - \hat{\ell}, r - \mathsf{c} - 1\}$, the invariants can be satisfied. Therefore, we can derive a lower bound on δ as follows:

$$\begin{aligned}
\delta \geq{} & \Pr[u_{1-b} \text{ does not send a share of challenge message}] \\
& \times\Big(1 - \Pr[u_{1-b} \text{ sends a message in } \{r-\hat{\ell}, r-\mathsf{c}-1\}] \\
& \quad - \Pr[u_{1-b} \text{ sends a message in } \{r-\mathsf{c}, r-1\}] \\
& \quad \times \Pr[\text{At least one of the } \mathsf{c} \text{ parties is honest}]\Big) \\
\geq{} & \left(1-\tfrac{B}{\mathsf{N}-1}\right)\left(1-\tfrac{B(\hat{\ell}-\mathsf{c})+(\hat{\ell}-\mathsf{c})}{\mathsf{N}} - \tfrac{B\mathsf{c}+\mathsf{c}}{\mathsf{N}} \times \left[1-1\Big/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\right)
\end{aligned}$$

Bound on anonymity when $B\hat{\ell} \leq \mathsf{N}$ and $\mathsf{c} < \hat{\ell}$. We can use a similar technique as above to derive a precise bound on δ when $B\hat{\ell} \leq 1$ and $\mathsf{c} < \hat{\ell}$. Since $B\hat{\ell} \leq \mathsf{N}$, for $\hat{\ell} < N$ the number of messages sent by bob is bounded by 2, and $\tau \leq 1$. Therefore, we can derive the following lower bound on δ:

$$\begin{aligned}
\delta \geq{} & \Pr[u_{1-b} \text{ does not send a share of challenge message}] \\
& \times\Big(1 - \Pr[u_{1-b} \text{ sends two messages in } \{r-\hat{\ell}, r-1\}] \\
& - \Pr[u_{1-b} \text{ sends only one message in } \{r-\hat{\ell}, r-\mathsf{c}-1\}] \\
& - \Pr[u_{1-b} \text{ sends only one message in } \{r-\mathsf{c}, r-1\}] \\
& \quad \times \Pr[\text{At least one of the } \mathsf{c} \text{ parties is honest}]\Big) \\
\geq{} & \left(1-\tfrac{B}{\mathsf{N}-1}\right)\Bigg(1 - max\left(0, \tfrac{B\hat{\ell}+\hat{\ell}-\mathsf{N}}{\mathsf{N}}\right) \\
& \quad - \tfrac{B(\hat{\ell}-\mathsf{c})+(\hat{\ell}-\mathsf{c})}{\mathsf{N}} - \tfrac{B\mathsf{c}+\mathsf{c}}{\mathsf{N}} \times \left[1-1\Big/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\Bigg)
\end{aligned}$$

Note that, $\Pr[\text{At least one of the } \mathsf{c} \text{ parties is honest}]$ can never be negligible. Because, even for $c = 1$, $1\Big/\binom{\mathsf{K}}{\mathsf{c}}$ is not negligible. The adversary can always choose to compromise less number of parties if that gives the adversary more advantage. This untightness is because of the approximations in our proof, tighter bounds are left for future work.

Therefore, a protocol can not achieve strong anonymity if $max\left(0, \frac{B\hat{\ell}+\hat{\ell}-\mathsf{N}}{\mathsf{N}}\right) + \frac{B(\hat{\ell}-\mathsf{c})+(\hat{\ell}-\mathsf{c})}{\mathsf{N}}$ is not overwhelmingly 1.

F.2 Calculating the probability of a specific user sending a message in a span of d rounds, for unsynchronized user message distribution

Here we derive an upper bound on the probability that a specific user (Bob) sends a message in a given span of d rounds, given that the protocol knows the frequency distribution of the messages over rounds. This approximates the cases where protocols can choose the delay of a message depending on the density of messages at different times. The calculations follow along the lines of the proof of Theorem 8 from [12].

Consider the following indicator random variables: $X^{(1)}, X^{(2)}, \ldots, X^{(\mathsf{N})}$, each $X^{(i)}$ denoting if user i sends a message or not in a span of d rounds. Since every user acts independent of all other users, $X^{(i)}$s are mutually independent, and $X^{(i)}$ can be defined as,

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

We further denote $X = \sum_{i=1}^{\mathsf{N}} X^{(i)}$. The expectation of X can be calculated as, $\mathbb{E}[X] = \sum_{i=1}^{\mathsf{N}} \mathbb{E}[X^{(i)}] = \mathsf{N}(1-(1-p)^d) = \mu$.

Using Markov's Inequality we can say, $\Pr[X \geq 2\mu] \leq \frac{1}{2}$.

At least one message is sent by our chosen user Bob is denoted by the event Y. If the total number of messages in the span of d rounds is $x \in \{0, \ldots, \mathsf{N}\}$, $\Pr[Y|X=x] \leq \frac{x}{\mathsf{N}}$.

For $2\mu \leq \mathsf{N}$, we can derive,

$$\begin{aligned}
& f_p^{SA}(d) = \Pr[Y] \\
={} & \Pr[X \geq 2\mu] \cdot \Pr[Y|X \geq 2\mu] + \Pr[X < 2\mu] \cdot \Pr[Y|X < 2\mu] \\
\leq{} & \Pr[X \geq 2\mu] \cdot \Pr[Y|X = \mathsf{N}] + \Pr[X < 2\mu] \cdot \Pr[Y|X = 2\mu] \\
\leq{} & \frac{1}{2} \cdot \frac{\mathsf{N}}{\mathsf{N}} + \left(1-\frac{1}{2}\right) \cdot \frac{2\mu}{\mathsf{N}} = \frac{1}{2} + \left(1-(1-p)^d\right).
\end{aligned}$$

If $2\mu > \mathsf{N}$, we get $f(d) = \Pr[Y] = 1$. Using Chernoff bound, we can derive derive a tighter bound $\Pr[X \geq 2\mu] = \sigma(d) \leq \exp\left(-2(\mu^2/\mathsf{N}^2)\mathsf{N}\right)$. However, since we are interested in impossibility results, and the difference is a constant factor $\frac{1}{2}$, we utilize the result obtained by using Markov's inequality. We formally define the probability of Bob sending a message in the given d rounds as, $f_p^{SA}(d) = min\left(1, \frac{1}{2} + \left(1-(1-p)^d\right)\right)$.

However, when we analyze the possibility of strong anonymity, if $pd \leq 1$ we use $f_p^{SA}(d) = \left(1-(1-p)^d\right)$ instead. Because $\sigma(d)$ becomes negligible in N, thus negligible in η.

Lemma 1. *When $pd \leq 1$, $\sigma(d) \leq \exp\left(-2(\mu^2/\mathsf{N}^2)\mathsf{N}\right)$ is negligible in N for $\mu = \mathsf{N}(1-(1-p)^d)$.*

Proof. For $pd \leq 1$ we can say,

$$\begin{aligned}
pd \leq 1 &\Rightarrow (1-p)^d < \frac{1}{e} \\
&\Rightarrow 1-(1-p)^d > \frac{1}{2} \\
&\Rightarrow \frac{\mu^2}{\mathsf{N}^2} > \frac{1}{4}
\end{aligned}$$

Therefore, $\sigma(d) \leq \exp\left(-2(\mu^2/\mathsf{N}^2)\mathsf{N}\right) \leq \exp(-\mathsf{N}/2)$ — which is always negligible in N. □

F.3 A tighter special case for Theorem 4

Let us derive a tighter upper bound on δ, in case of unsynchronized user message distribution, when $0 \leq \mathsf{c} < \ell$. W is a random variable denoting the minimum number of paths that the adversary needs to compromise to ensure no honest party on the paths of the shares of the challenge messages as well as no honest party on the paths of the messages from u_{1-b}. When $\mathsf{c} < \hat{\ell}$,

$$\begin{aligned}
&\Pr\left[\text{Invariant 2 is true}\right] \\
\leq &\Pr\left[u_{1-b} \text{ sends a share of the challenge message.}\right] \\
+&Pr[u_{1-b} \text{ sends no shares of the challenge message}] \\
\times &\Bigg(Pr[u_{1-b} \text{ sends a message in } \{r-\hat{\ell}, r-\mathsf{c}-1\}] \\
+&Pr[u_{1-b} \text{ does not send a message in } \{r-\hat{\ell}, r-\mathsf{c}-1\}] \\
\times &\Big(Pr[u_{1-b} \text{ sends more than one message in } \{r-\mathsf{c}, r-1\}] \\
+&Pr[u_{1-b} \text{ sends only one message in } \{r-\mathsf{c}, r-1\}] \\
\times &Pr[\text{Some share of the challenge message visits honest} \\
&\quad \text{node and some message from } u_{1-b} \text{ visits honest node}]\Big)\Bigg) \\
\leq &\frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\Bigg[f_p^{SA}\left(\hat{\ell}-\mathsf{c}\right) + \left(1 - f_p^{SA}(\hat{\ell}-\mathsf{c})\right) \\
&\quad \times \Bigg(\Pr\left[W \geq 2 \wedge Y(\mathsf{c}) \geq 2\right] \\
&\quad + \Pr\left[W = 1 \wedge Y(\mathsf{c}) \geq 1\right] \times \left[1 - 1/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\Bigg)\Bigg] \\
\leq &\frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\Bigg[f_p^{SA}(\hat{\ell}-\mathsf{c}) + \left(1 - f_p^{SA}(\hat{\ell}-\mathsf{c})\right) \\
&\quad \times f_p^{SA}(\mathsf{c})\Bigg(\Pr\left[W \geq 2\right] + \Pr\left[W = 1\right]\left[1 - 1/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\Bigg)\Bigg]
\end{aligned}$$

Note that W is a random variable, where $W = min\left(\frac{(X-X')}{X'} + 1, X\right)$. Here X and X' follow $Binom(\mathsf{c}, p)$ and $Binom(\mathsf{c}, p')$ respectively. Therefore, We can say that $Pr[W = 1]$ is bounded by $Pr[W = 1] \leq w_1 = Pr[X = 1] = \mathsf{c}p\,(1-p)^{\mathsf{c}-1}$. Consequently, $Pr[W > 1] \geq w_2 = Pr[X > 1] = 1 - w_1 - (1-p)^{\mathsf{c}}$. Therefore, we can write,

$$\begin{aligned}
&\Pr\left[\text{Invariant 2 is true}\right] \\
&\leq \frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\Bigg[f_p^{SA}(\hat{\ell}-\mathsf{c}) + \left(1 - f_p^{SA}(\hat{\ell}-\mathsf{c})\right) \\
&\quad \times f_p^{SA}(\mathsf{c})\Bigg(w_2 + w_1\left[1 - 1/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\Bigg)\Bigg]
\end{aligned}$$

Thus, $\delta \geq 1 - \Pr\left[\text{Invariant 2 is true}\right]$

$$\begin{aligned}
&\geq \left(1 - \tfrac{B_{\text{eff}}}{\mathsf{N}-1}\right) \times \left(1 - f_p^{SA}(\hat{\ell}-\mathsf{c})\right)\Bigg[1 - f_p^{SA}\left(\mathsf{c}\right) \\
&\quad \times \Bigg(w_2 + w_1\left[1 - 1/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\Bigg)\Bigg].
\end{aligned}$$

F.4 Analyze average case of the user distribution, to derive impossibility conditions for strong/quadratic anonymity

Lemma 2. *Let R be the set of all possible* sequences of execution *of an AC protocol. Let $Runs \in R$ be a random variable denoting the sequence of execution. Suppose, for a set of sequences of execution $R' \subset R$,* $\Pr[Runs \in R']$ *is μ, and μ non-negligible. If the protocol can not provide strong anonymity for any execution $o \in R$, the protocol can not provide strong anonymity overall.*

Proof. Suppose Y denotes the event that the adversary wins, and o^* is the element in R' for which the probability that the adversary wins is maximum, i.e., $\Pr\left[Y \mid o^*\right] \geq \Pr\left[Y \mid o\right]$ for all $o \in R'$. Suppose, $\Pr\left[Y \mid o^*\right] = \nu$. Then we can say,

$$\begin{aligned}
\Pr[Y] &= \sum_{o \in R} \Pr[Y \mid Runs = o] \cdot \Pr[Runs = o] \\
&= \sum_{o \in R'} \Pr[Y \mid Runs = o] \cdot \Pr[Runs = o] \\
&\quad + \sum_{o \in R \setminus R'} \Pr[Y \mid Runs = o] \cdot \Pr[Runs = o] \\
&\leq \sum_{o \in R'} \Pr[Y \wedge Runs = o^*] \cdot \Pr[Runs = o] + S_{R''} \\
&= \Pr[Y \wedge Runs = o^*] \cdot \Pr[Runs \in R'] + S_{R''} \\
&= \nu \cdot \mu + S_{R''}
\end{aligned}$$

where $S_{R''} = \sum_{o \in R \setminus R'} \Pr[Y \mid Runs = o] \cdot \Pr[Runs = o]$. We know μ and ν both are non-negligible. Therefore, $\Pr[Y]$ is non-negligible. □

The above lemma provides us with a very helpful insight for unsynchronized user message distribution. Suppose X denotes the total number of messages in a given slice of $\hat{\ell}$ rounds, and R' denotes the set of all sequences of execution where $X < \mathbb{E}[X]$. For two values x_1 and x_2 drawn from X and $x_1 > x_2$, the protocol has a better chance for anonymity with $X = x_1$. Therefore, if we are analyzing the possibility of strong anonymity, it is enough to analyze the average case scenario in most of the cases (e.g., $p\hat{\ell} \in O(1)$ – in which case we can replace $f_p^{SA}(d)$ with $\left(1-(1-p)^d\right)$ for a given slice of d rounds). Additionally, we can use $\delta \geq \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\left[1 - g(Z) \times f_p^{SA}(\hat{\ell})\right]$, where $Z = min\left(\hat{\ell}, 2B+1\right)$ to analyze strong anonymity. Because, by Markov inequality, the number of additional shares for a message will be bounded by $2B$ with probability at least $\frac{1}{2}$.