Anonymity Trilemma: Beyond Mix-Nets

Debajyoti Das
Purdue University, USA
das48@purdue.edu

Sebastian Meiser
Visa Research, USA
smeiser@visa.com

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

Aniket Kate
Purdue University, USA
aniket@purdue.edu

Abstract

In this work, we identify a class of Anonymous communication (AC) protocols that can achieve a higher level of anonymity compared to mix-nets, by having out-of-band coordination among users in addition to all the tricks that mix-nets can use. We argue that this new class of protocols can achieve the best of two worlds, mix-nets and DC-net-type protocols, and characterizes most of the commonly known protocols. We analyze the upper bound on anonymity for this wider class of protocols as a function of latency overhead and bandwidth overhead, against global passive adversaries that can additionally passively compromise some of the protocol parties.

Even for this wider class of protocols, we prove the widely believed trilemma: It is impossible to achieve all of the following three properties — strong anonymity, low latency, low bandwidth overhead. As a byproduct, we derive the specific impossibility conditions for strong anonymity; and we show that, the impossibility conditions are much relaxed for our new class of protocols compared to mix-nets. Additionally, we show that, these protocols have a much better chance at anonymity compared to mix-nets, if strong anonymity is not required, given a restriction on latency and bandwidth overhead. Most importantly, by proving the impossibility conditions of anonymity for this wider class of protocols, we show that the only scope to escape the widely believed trilemma is to escape the wide class of protocols that we characterize in this work; and that is only possible through expensive crypto-magic like fully homomorphic onion encryption, efficient amortized secret sharing etc. This guides the protocol designers towards a new direction of research for AC protocols.

CONTENTS

I. Introduction

Anonymous communication networks (ACNs) are critical to communication privacy over the Internet. After almost four decades of work, the search for an optimal ACN design is still ongoing as we continue to find an ACN that has minimal latency and bandwidth overhead, provides strong anonymity, and is maximally robust against compromised protocol parties.

Recent work [1], [2] has reduced the search space of possible ACN designs by proving necessary constraints (i.e., impossibility results) for ACNs that relate bandwidth overhead, latency overhead, the degree of anonymity, and robustness to compromised protocol parties (e.g., mix-net nodes). These existing necessary bounds leave very little hope for low-latency ACNs in the presence of compromised protocol parties for mix-nets: in strongly anonymous mix-nets, no matter the bandwidth overhead, messages cannot have short latency (i.e., constant w.r.t. to some security parameter) in the presence a large number of compromised parties. In fact, mix-nets from the literature [3]–[5] propose to use very long latency to ensure robustness against compromised parties (compromised mixes).

Fortunately, this necessary constraint on the latency seems to be particular to mix-nets. DC-nets and its successors [6], [7], e.g., are able to achieve very short latency (1 round) at the cost of sending for each message one additional message to each client (or to each client in a subgroup). Moreover, recent ACN proposals [8] propose to use dedicated computation servers and to run modern cryptographic multi-party protocols (e.g., variants of private-information retrieval) to achieve resistance to compromised parties without requiring a very long latency. This line of work naturally raises the following question:

> How much bandwidth overhead is necessary to achieve strong anonymity if low latency is desired in the presence of compromised parties?

Generalizing the techniques from this line of work, our work identifies a property that is reminiscent of secret-sharing protocols: assuming some form of synchronization among protocol protocol parties, even the recipients of a message cannot distinguish the packet sent by the real sender and the dummy messages. As an example, secret-sharing a real message offline among several protocol parties and then sending the shares at a pre-agreed time, would achieve this property, since the content of the message would be shared among all messages and would not solely be contained in the message of the real sender. Alternatively, DC-nets shared keys to produce dummy messages that are needed to reconstruct the original message, which also achieves our proposed property.

We abstract away from how ACNs would synchronize and solely analyze the *core of a secret-sharing based ACN*: for this core we prove a necessary relationship between bandwidth, latency, and the degree of anonymity in the presence of compromised parties. In particular, for ACNs that can perform the above-mentioned synchronization offline, our novel necessary constraints give lower bounds on the bandwidth overhead for ACNs with low latency and strong anonymity in the presence of compromised parties. Our results thereby points future research on designing optimal ACNs in the direction of secret-sharing based ACNs.

A. Our Contribution

We identify a class of protocols that we coin secret-sharing based ACNs, which require a synchronization procedure. Secret-sharing protocols have to ensure that even the recipients of a message cannot distinguish the packet sent by the real sender and the dummy messages.

We show that the core of these protocols (i.e., with the synchronization procedure) can escape the necessary constraint on low latency for mix-nets with strong anonymity in the presence of compromised parties. In particular, if the ACN can perform the synchronization procedure offline, the ACN can escape the prior necessary constraints on low latency.

We derive new upper bound on anonymity for these protocols, and deduce new necessary constraints for strong anonymity. We additionally analyze the possibility of a weak version of anonymity in those cases. In the process of deriving bounds for our new wider class of protocols, we also improve upon the existing bounds for mix-net type protocols.

We discuss the tightness of our novel necessary constraints, and analyze interesting cases. We furthermore discuss whether prior literature on ACNs realizes the property on ACNs that we are proposing.

Finally, our results should make the recent ACNs designers to reconsider their protocols. While building scalable strong anonymity ACN solutions, they have continued to stick to layered encryptions for their cryptographic protocol choice. Our results will motivate the ACN dedigners to reconsider their choice and employ secret-sharing approach for lower latency and bandwidth overhead.

II. Overview

We identify two classes of protocols: (1) *classical protocols* that do not use any kind of secret sharing technique, (2) *secret-sharing protocols* that use some kind of secret sharing technique. The work by Das. et. al. models only the protocols what we call classical protocols. these protocols do not use any kind of secret sharing techniques, i.e., as they write, "*in order for Bob to receive a message from Alice, Alice has to send the message and the message (albeit relayed, delayed and modified) eventually has to reach Bob*". In this work, we model a wider class of protocols, where users can also collude among themselves (out-of-band) to send shares of a message to the recipient, and the recipient can retrieve the message only after receiving all the shares.

Communication Rounds, Latency and Bandwidth Overhead. Protocols in our model follow synchronous rounds for communication as in [9]–[12]. Every protocol is constrained by two important parameters: the *latency overhead ℓ*, which describes how many rounds each message may reside in the protocol before it has to be delivered and the *bandwidth overhead B* describing the number of noise messages the protocol can create for each real message.

Adversaries. Following [13], we consider global passive adversaries, that observe all communication between protocol parties and that can additionally compromise c protocol parties to learn the mapping between inputs and outputs for this party. Note that a compromised party follows the protocol specifications correctly, and can not learn anything other than the mapping between inputs and outputs for that party.

Anonymity Property. We use an indistinguishability-based anonymity notion to quantify sender anonymity [14]–[17]: we measure the adversarial advantage as the probability that the adversary can distinguish two senders of its own choosing; anonymity then describes limits on this adversarial advantage.

Given a security parameter η that can be related to the number of protocol parties, we say that a protocol provides *strong anonymity* in η, or *strong anonymity* for short, if the adversary's advantage δ is negligible in η. In Section III we explain the significance of such a security parameter in detail.

A. A primer on impossibility trilemma proofs

To provide bounds for secret-sharing protocols, we follow a proof technique similar to that of Das. et. al. [13] We derive the lower bounds on delta, alternatively upper bound on anonymity, in four steps as described below.

The first step is to define a concrete adversary $\mathcal{A}_{paths}$ is the passive (but can passively compromise some protocol parties) adversary class. Any advantage gained by the $\mathcal{A}_{paths}$ is at least as much as the advantage gained by the strongest adversary in the given adversary class.

In the second step, we identify an invariant, if remain unfulfilled by a protocol, will ensure that $\mathcal{A}_{paths}$ wins. And therefore, a protocol can at best achieve as much anonymity as the probability of satisfying the invariant, against the given (passive) adversary class.

Third, we identify an *optimal protocol* Π_{ideal} that satisfies the invariant with a probability at least as high as any other other protocol in our model (bounded by the same ℓ and B restrictions).

Finally, we calculate an upper bound on the probability of Π_{ideal} fulfilling the necessary invariant against $\mathcal{A}_{paths}$ - which gives us a lower bound on the adversarial advantage δ for the passive adversary class against all protocols in our model.

B. Advantages of using secret sharing in AC protocols

When protocols are allowed to use more than onion routing, the protocol will not gain any additional advantage from that in case all the intermediate protocol parties are honest, and hence, the bounds will remain the same as Das. et. al. [13] However, when the adversary is allowed to compromise protocol parties, secret sharing techniques can provide resistance against that, and provide the protocol some advantage.

To explain the advantage gained by a protocol with secret sharing, let us consider an example with $\ell = 0, B = \mathsf{N}$, in the case of synchronized user distribution. Since $\ell = 0$, a protocol does not have any chance of mixing the messages using an intermediate party, users will have to directly dispatch the messages to the recipient. Without any secret sharing technique, the protocol can not achieve any anonymity. On the contrary, if a protocol can use out-of-band secret sharing technique, all N users can send a share each for a real message. When the recipient receives all the shares, he has to combine them to retrieve the actual message. If the shares themselves do not reveal anything about the sender, the recipient has no way to determine who among the N users is the real sender — Thus, the protocol can achieve anonymity. Although this is an extreme example where $\ell = 0, B = \mathsf{N}$, it demonstrates the fact that secret sharing can help a protocol.

In another example, when $\frac{\mathsf{K}}{\mathsf{c}} \geq 2, B < \frac{\mathsf{N}}{2log(\eta)}, \ell < log(\eta)$, neither classical nor secret sharing protocols can achieve strong anonymity as shown in Table III in Section IX. However, the advantage of the adversary can be much lower against a secret sharing protocol compared to a classical protocol.

More generally, we show that there are inherent lower bounds on δ, depending on the number of users N, the latency overhead ℓ, the bandwidth overhead B and the number of compromised parties c out of the K (internal) protocol parties:

Synchronized Users:

$$\delta \geq \left(1 - \tfrac{B}{\mathsf{N}-1}\right) \times \Big[1 - \tfrac{(\tau+1)\mathsf{N} - B(\ell+1) - (\ell+1)}{\mathsf{N}} \times g(\tau) - \tfrac{B(\ell+1)+(\ell+1)-\tau\mathsf{N}}{\mathsf{N}} \times g(\tau+1)\Big]$$

where $\tau = \lfloor \frac{B(\ell+1)+(\ell+1)}{\mathsf{N}} \rfloor$, and

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

Unsynchronized Users:

$$\delta \geq \begin{cases} \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right) \left[1 - g(Z) \times \left(1 - (1-p)^{\ell+1}\right)\right], & \mathsf{c} \geq \ell+1 \\ \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right) (1-p)^{\ell+1-\mathsf{c}} \Big[1 - \left(1 - (1-p)^{\mathsf{c}}\right) & \\ \times \left(\Pr[W \geq 1] + \Pr[W = 0] \left[1 - 1/\binom{\mathsf{K}}{\mathsf{c}}\right]\right)\Big] & \mathsf{c} < \ell+1 \end{cases}$$

where $B_{\text{eff}} = min(B, (\ell+1)p-1)$, $Z = min(\ell+1, 2B_{\text{eff}}+1)$, W is a random variable denoting the number of additional shares for the challenge message.

C. Related Work

Das et al. [13] formalize and confirm the anonymity trilemmna. They provided the formally bounded necessary conditions for anonymity and showed for which parameters of bandwith overhead and latency overhead *strong anonymity*, i.e., anonymity up to a negligible adversarial advantage, is impossible. However, in the presence of compromised protocol parties, that work solely provides necessary constraints for mix-nets, not for secret-sharing based ACNs.

In a prior work, Oya et al. [18], also provides a generic attacker in a general model that encompasses a large class of ACNs. That work, however, concentrates 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, latency overhead, the compromisation rate and the degree of anonymity.

Recently, Ando et al. [19] derived necessary constraints for communication complexity and the degree of anonymity in the presence of active attackers for mix-nets. Hence, that work does not capture bandwidth overhead and, more importantly, does not provide necessary constraints for secret-sharing based protocols. More recently, Ando et al. [20] proved about mix-nets that anonymity can only be achieved if each client transmits on average a superlogarithmic number of packets.

III. The anonymity game

We define anonymity by a game between a challenger (controlling the protocol) and a global passive adversary, following the AnoA framework [16], [17]. 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, e.g., for sender anonymity, 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. Formally, the anonymity definition is as follows:

Definition 1 ((α, δ)-IND-ANO from [13])**.** *A protocol* Π *is* (α, δ)-IND-ANO *for the security parameter* η*, an adversary class* C*, an anonymity function* α *and a distinguishing factor* $\delta(\cdot) \geq 0$*, if for all PPT machines* $\mathcal{A} \in$ C*,*
$\Pr[0 = \langle \mathcal{A} | \mathtt{Ch}(\Pi, \alpha, 0) \rangle] \leq \Pr[0 = \langle \mathcal{A} | \mathtt{Ch}(\Pi, \alpha, 1) \rangle] + \delta(\eta)$.

Π *provides* strong α-anonymity *[13]–[15] if it is* $(\alpha, \delta) -$ IND-ANO *with* $\delta \leq neg(\eta)$ *for some negligible function neg.*

Π *provides* quadratic α-anonymity *if it is* $(\alpha, \delta) -$ IND-ANO *with* $\delta \in O(\frac{1}{\eta^2})$.

Here $\langle \mathcal{A} | \mathtt{Ch}(\Pi, \alpha, b) \rangle$ stands for the interactive game between the 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.
- $(\mathtt{Chall}, u_0, u_1, R, m)$ for sender anonymity, in which case the challenger selects one user based on the challenge bit b, and then instructs user u_b send a message m to recipient R.

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

On the meaning of η. In our analyses we tie η to system parameters such as P, [c], ℓ, B, 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 composition (e.g., for streams of messages or usage over a longer time period) and formally, η limits the number of compositions.

A. What can and cannot protocols do?

Anonymous communication protocols are still 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.

B. Adversary

Following [13], 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 protocol parties.

C. User message distributions

We follow [13] in their distinction between two types of *user distributions*, i.e., two different definitions of hoe 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 particularly protocol friendly and over-approximates many ways that the protocol could in an offline-phase synchronize when users send messages. A similar kind of synchronization is actually even discussed for DC-net kind of protocols (to ensure that messages from a sender can actually be reconstructed). Our results show that even for this protocol-friendly user distribution U_B many interesting cases are the same as for the unsynchronized U_P.

IV. A Protocol Model for AC Protocols

We follow [13] in our definition of a protocol model but extend a protocol's capabilities by allowing some secret-sharing style techniques. Technically, a protocol is defined as a colored Petri net in which senders $\mathcal{S}$ send packets to recipients $\mathcal{R}$ via some anonymizing proxies P. Protocols operate in rounds. Whenever a packet is sent from one of these entities to another, the eavesdropping adversary learns that a packet is sent as well as the round in which this occurred. 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. In addition, the adversary compromises all recipients and upon receiving packets can learn their content. Secret-sharing techniques add to the adversary's confusion here by requiring receiving several packets to (re-)construct a real message.

In this section we introduce the necessary formalism for understanding our results and describe how we modify the original Petri net model. For technical details we refer to [13].

A. Protocol Model

An AC protocol M consists of places $\mathcal{S}$ for senders, $P_1, \ldots, P_k$ for proxies/internal protocol parties and places $\mathcal{R}$ for recipients of packets as well as an additional place \$ populated with random coins. Each packet is a colored token (read: tuple of values) that in every round is allowed to transition from one place to another (using a transition T). Each packet $q = \langle \mathsf{tag},\ \mathsf{meta},\ \mathsf{t_r},\ \mathsf{ID_t},\ \mathsf{prev},\ \mathsf{next},\ \mathsf{ts} \rangle$ is comprised of four public fields ($\mathsf{ID_t}$, prev, next, ts) that can be observed by the adversary and three private fields (tag, meta, $\mathsf{t_r}$) that are hidden from the adversary with the exception that the field tag is revealed to the recipient.[1]

- tag represents the content of the packet that the recipient will use to reconstruct a message,
- meta contains all internal protocol meta-data for this message (and is not important for our analysis),
- $\mathsf{t_r}$ is the time (in number of rounds) the message can still remain in the network (initially this is set to ℓ),
- $\mathsf{ID_t}$ 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 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 net). Here, ts either describes when new messages are introduced into the Petri net or is set to the next round, such that messages can be processed in every round as soon as they enter the network.

The recipient has access to a dictionary D (outside the petri-net); when a packet reaches the recipient, the recipient queries a dictionary D to retrieve the corresponding 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.

```
$\mathbf{T_X}$ on tokens $q = \langle \mathsf{tag}, \_, \mathsf{t_r}, \mathsf{ID_t}, \_, \mathsf{prev}, \mathsf{ts}, \mathsf{tag} \rangle$ 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 $P' = R$ then obs $= \langle \mathsf{tag}, \_, \_, \mathsf{ID_t}', \mathsf{prev}, P', 1 \rangle$
    else obs $= \langle \_, \_, \_, \mathsf{ID_t}', \mathsf{prev}, P', 1 \rangle$
    $\mathsf{Tokens} = \mathsf{Tokens} \cup \{(\mathrm{obs}, r)\}$
Output: token $t$ at $P'$

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

Reconstruct(tag):

    if $\mathsf{tag} = \perp$ or $\mathsf{D}[\mathsf{tag}]$ does not exist then return $\perp$
    $\mathsf{D}[\mathsf{tag}].\mathtt{count} = \mathsf{D}[\mathsf{tag}].\mathtt{count} + 1$
    if $\mathsf{D}[\mathsf{tag}].\mathtt{count} = \mathtt{count.countNeeded}$ then return $\mathsf{D}[\mathsf{tag}]$
    else return _
```

Fig. 1. Transitions in the Petri net model M

Transitions. A Petri net uses transitions to move tokens from one place to another, to modify them and to consume or create tokens. In our case, we mostly move tokens from one place to the next which represents the transmission of a packet from one protocol party to another. In the process of moving a token, we change their field $\mathsf{ID_t}$ whenever they are moved out of the place of an honest proxy, which simulates that the adversary cannot trace packets through an honest proxy. From a technical point of view, transitions implement the representation of the network and provide the adversary with the appropriate observations. A protocol cannot influence what each transition does, but can choose which transition is taken, with the important exception that tokens that run out of their allowed latency overhead always are sent to the recipient.

Initially, all places except for the $\mathcal{S}$ place and the place for randomness \$ are empty. Transitions consume tokens from those two places to place tokens in other places, which represents the movement of packets.

We have only one type of transaction that moves tokens from one place to another, representing the transmission of a packet from one entity to another. Each transaction T_X takes a token q from the place $X \in \mathcal{S} \cup \mathsf{P}$ and creates two new tokens:

[1] Since we only consider sender anonymity we do not need to specify the recipient in the token.

(1) a token q' representing the same packet that is handed to a new place and (2) an adversarial observation (obs, r) annotated with the round number r that is added to a set of observed tokens Tokens. We refer to Figure 1 for a technical description of the algorithm implemented by T_X.

Validity of M. The way of populating $\mathcal{S}$ and defining transitions naturally enforces bandwidth and latency overheads. Our model stays close to that of [13] and thus leverages their validity proof. Our definitions merely simplify their formalism and thus don't affect the validity on a per-packet basis. We do add on message tags and a reconstruction algorithm, but sufficiently restrict it when populating $\mathcal{S}$.

V. Impossibility bounds for mix-nets

The work by Das. et. al. [13] takes a first step at deriving impossibility results for ACNs, but they only consider mix-net protocols where users do not use any secret sharing. Which means, "in order for Bob to receive a message from Alice, Alice has to send the message and the message (albeit relayed, delayed and cryptographically modified) eventually has to reach Bob". This rules out protocols like DC-net or other secret sharing based protocols. In this section, we are going to summarize the bounds on anonymity for mix-net presented by Das. et. al. [13].

Necessary invariant for protocol anonymity. It is necessary that at least both challenge users send messages in one of the ℓ rounds before the challenge message reaches the recipient, as otherwise there is no way both of them could have sent the challenge message. Moreover, on the path of the actual challenge message, there needs to be at least one honest (uncompromised) party, as otherwise the adversary can track the challenge message from the sender to the recipient. Those two conditions together form the *necessary protocol invariant* in the work by Das. et. al.

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

(i) the set T is not empty, and

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

A. Lower Bounds on adversarial advantage

Using Invariant 1, Das. et. al. derive the following lower bounds on the adversarial advantage δ against mix-net type protocols for synchronized (U_B) and unsynchronized (U_P) user distributions.

Theorem 1. *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 2. *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 $f_\beta(x) = \min(1, ((x + \beta \mathsf{N} x)/(\mathsf{N} - 1)))$.

Theorem 3. *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 4. *For user distribution U_P, no protocol $\Pi \in M$ can provide δ-sender anonymity, for any*

$$\delta < \begin{cases} 1 - [1 - \binom{\mathsf{c}}{\ell}/\binom{\mathsf{K}}{\ell}][\frac{1}{2} + f_p(\ell)] & \mathsf{c} \geq \ell \\ \Big(1 - [1 - 1/\binom{\mathsf{K}}{\mathsf{c}}][\frac{1}{2} + f_p(\mathsf{c})]\Big) & \\ \qquad \times \Big(1 - [{}^1\!/\!_2 + f_p(\ell - \mathsf{c})]\Big) & \mathsf{c} < \ell \end{cases}$$

where $f_p(x) = \min({}^1\!/\!_2,\ 1 - (1-p)^x)$ for a positive integer x.

Throughout our paper we shall call them *classical impossibility bounds*. The impossibility conditions provided by the classical bounds are summarized in Table I.

TABLE I
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 $\ell \in \mathcal{O}(1)$

VI. Impossibility bounds beyond mix-nets

Here we will constructively investigate an abstract protocol within our model that combines secret-sharing with mixing techniques. We then show that, indeed, the protocol can achieve a better degree of anonymity than the classical impossibility results in Section V indicate.

The intuitive reason for this effect is that such protocols can introduce ambiguity in terms of which message content is within which network packet. Imagine an adversary that compromises every node in the path that a particular packet traverses and that then observes 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 respective message sent in the particular round and thus all potential senders of these packets could be the sender of the message.

A. An AC Protocol Involving Secret Sharing:

Our envisioned protocol falls within our protocol model (Section IV) and, crucially, leverages secret-sharing techniques as follows: Users that don't want to send messages in a given round can send noise messages of a special kind that we call *shares*. Each such share is associated with one real message (with content) within the system and the recipient needs to collect all the shares for a message in order to decipher it. When a message and its shares 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 secret sharing and hence can not decipher an individual secret before it reaches the recipient. Additionally, we assume an efficient out-of-band secret sharing. (For instance, in DC-net [21] with pre-setup key agreement, the protocol parties only need to publish their local messages.)

The protocol works in the following way:

1) Based on the user distribution, users decide when to send messages.

2) Whenever a user is supposed to send a noise message, he just participates in a secret sharing for a real message from some other user. Instead of a noise, the user sends a share.

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

4) We assume a series of relays (up to K relays), out of which c (chosen uniformly at random) are compromised; and using those relays the users can send the messages to the recipient. Once the protocol starts, the sequence of the relays is known to all users. Suppose, Alice is supposed to send a message at round t, and her message is supposed to be delivered at round r. Alice *onion-encrypts* the message to ensure it is delivered at round r.

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

Analysis of Adversarial Advantage for the above protocol. We know from Section V that for the *synchronized user distribution*, the adversarial advantage δ should be lower bounded by, $\delta \geq 1 - \left[1 - \binom{\mathsf{c}}{\ell}/\binom{\mathsf{K}}{\ell}\right] \times min\left(1, \frac{\ell+\beta\mathsf{N}\ell}{\mathsf{N}-1}\right)$.

However, for our described protocol, if by chance the user u_{1-b} is sending any of the shares of the challenge message, the adversary can not win, even if it can trace all messages and arbitrarily many nodes are compromised. Therefore, $\delta \leq 1 - \frac{\beta\mathsf{N}}{\mathsf{N}-1}$. Recall that the consensus and secret sharing can happen out of band and thus doesn't add any bandwidth overhead here.

Hence, our protocol escapes the impossibility result in [13], for $\ell = 1, \mathsf{K} = 2, \mathsf{c} = 1$, when the following is true:

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

$$\Longleftrightarrow \frac{\beta\mathsf{N}}{\mathsf{N}-1} > \frac{1}{2} \times min\left(1, \frac{\ell+\beta\mathsf{N}\ell}{\mathsf{N}-1}\right)$$

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

$$\Longleftrightarrow 2\beta\mathsf{N} > 1 + \beta\mathsf{N} \qquad \because \ell = 1$$

$$\Longleftrightarrow \beta\mathsf{N} > 1 \iff \beta > \frac{1}{\mathsf{N}}$$

Thus, even if only one user per round can send one share, our protocol violates the classical impossibility bounds. We now proceed to analyze the effect that such secret sharing can have on anonymity before deriving novel impossibility results that are still valid in light of secret sharing.

B. Effect of Secret Sharing on Anonymity

As we demonstrated above, an AC protocol can utilize secret sharing to increase the chance of achieving anonymity. If a set of users sends shares for a given message, the adversary has no way to distinguish the actual sender of the message from other users in the set, unless the secret sharing scheme is broken. This feature of secret sharing provides the AC protocols with a resistance against compromised relays. If an AC protocol can achieve an efficient/out-of-band secret sharing scheme (like DC-net) as well as can utilize relay nodes, it can escape the impossibility results from Section V. However, if all the relay nodes are honest, secret sharing does not provide any additional advantage, and the impossibility results from Section V remain valid. In the following sections, we shall present new lower bound on adversarial advantage (upper bound on anonymity) in the presence of compromised relay nodes, both for the synchronized user distribution and for the unsynchronized user distribution.

We use the protocol model, adversarial model, and security game as described in Section IV to derive our impossibility bounds. In this section, we define new necessary invariant, more relevant for secret sharing based protocols, against our well defined adversary $\mathcal{A}_{paths}$. Then we define an ideal protocol which maximizes the probability of satisfying the invariant against $\mathcal{A}_{paths}$. In the later sections, we follow the proof technique described in Section II to derive lower bounds on adversarial advantage. But first, we formally state the assumptions on secret sharing based protocols in our model.

C. Assumption on Secret Sharing Protocols

Secret sharing techniques span a wide array of actual methods, some of which we deem unreasonably strong. In this section we highlight which protocols our protocol model intentionally excludes:

Challenge 1. *Our necessary constraints are based on the assumption that the protocol does not use a secret sharing scheme that generates $w < k \times n$ shares for n messages where at least k shares are necessary to reconstruct each of the m messages correctly, without using any trusted third party, with* *a communication of $O(m)$ and constant latency overhead. If a protocol used such a scheme it would be possible for that protocol to achieve strong anonymity with constant latency overhead, and constant bandwidth overhead per message.*

Invisible senders: We also assume that one of the users who send the shares is the actual sender. We do not consider the scenario where a set of users send shares for user u, however u is not part of that set. In that case, user u will have to explicitly distribute the shares to the other users directly or through a trusted third party. Then, that can be considered as the "Splitting and recombining" scenario mentioned in **??** and thus follows our previous impossiblity bounds.

Cheating the latency bound with shares: To conform with the latency overhead ℓ, if a message is scheduled to be sent in round t_0 by the user distribution, all shares of that message must reach the recipient before round $t_0 + \ell$.

Expensive cryptographic techniques: If the protocol uses strong and expensive cryptographic primitives, such as some form of obfuscation / homomorphic outsourced computation / non-interactive multi-user ORAM property[2], then the following becomes possible: even passively compromised nodes can be used to mix messages, i.e., the node cannot know which of its own outputs are related to which of its own inputs.

Consider the following instructive example, where two users, Alice and Bob send messages to a node M and utilize a variant of functional encryption. Alice sends: $Enc_{\mathcal{F}}[x, r]$ and Bob sends: $Enc_{\mathcal{F}}[y, r]$, where r is the current round number.

M now computes $\mathcal{F}(Enc[x, r], Enc[y, r']) := \text{sorted}(x, y)$ if $r = r' \land x \neq y$, and thus, M learns both x and y, but cannot know which of them came from which user. M cannot decrypt only one of them. We acknowledge that this is not necessarily an efficient or realistic instantiation, which is why we chose not to model this or similar behavior.

Consequence: No-obfuscation assumption: A compromised node can always relate incoming and outgoing packets. More precisely, if two packets enter a node and two packets leave a node, the node is able to tell how much information of each incoming packet is encoded in which outgoing packet.

Challenge 2. *Our impossibility result is based on the assumption that no protocol can achieve mixing in a dishonest node. We leave it as an open question whether an efficient method for doing so exists.*

Why we don't consider packet splitting and recombining at a node level. Without obfuscation (which could allow compromised nodes to mix messages), nodes could still split and recombine messages. We here assume that messages are already maximally densely packed, so any splitting and recombination does not reduce the size of transferred information. [3] However, note that any splitting at a node-level could be done at the client level as well. The clients can already send divided messages (we then count additional parts of a message as bandwidth overhead). This only shows that we can ignore node-level message-splitting, which we could not easily account for.

D. The path possibility adversary

Formally, we utilize a path possibility adversary $\mathcal{A}_{paths}$ as in the work of Das. et. al. [13]: 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 construe 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 obviously 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 please see Section A-A.

E. Necessary invariant for protocol anonymity

To prove our version of the anonymity trilemma for protocols with secret sharing, we define a necessary invariant, i.e., if the invariant is not satisfied for a protocol run, then even a fairly simple adversary will win independently of any further actions taken by the protocol.

Analogously to Section V we now derive new an invariant that remain necessary for anonymity and are more relevant in the presence of secret-sharing techniques.

Invariant 2 (New Invariant)**.** *Let u_0 and u_1 be the challenge users; let b be the challenge bit; and let t_0 be the time when u_b sends the challenge message. Assume that the challenge message reaches the recipient at r. Assume furthermore that u_{1-b} sends her messages (including noise messages) at $V = \{t_1, t_2, t_3, \dots, t_k\}$. Now, let $T = \{t : t \in V \land (r - \ell) \leq t < r\}$. Then,*

(i) the set T is not empty, AND

(ii) a. at least one share of the challenge message is dispatched by u_{1-b} within rounds $\{(r - \ell), \dots, (r - 1)\}$, 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 $t \in \{(r - \ell), \dots, (r - 1)\}$, 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 β. Let u_0, u_1, b and T be defined as in* **??**. *If Invariant 2 is not satisfied by Π, then our adversary $\mathcal{A}_{paths}$ as in Section VI-D wins.*

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

Whenever part(i) is false, the set T is empty. As a result, the adversary certainly knows that the challenge message is not sent by the u_{1-b}.

[2] The Pung protocol [22], goes in this direction by using PIR. The Pung protocol is, however, very inefficient.

[3] We assume that encryption is free.

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

However, given that part(ii.a) is already false, part(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 challenge message starting from the recipient, the path will never cross the paths of the messages from u_{1-b} at an honest node. So, the adversary will be able to exclude all of them from the possibility of being a share of the challenge message. That way, $\mathcal{A}_{paths}$ can eliminate u_{1-b} from the possibility of being the challenge user, 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 challenge message starting from the recipient, the path will never cross the paths of the messages from u_{1-b} at an honest node. So, the adversary will be able to exclude all of them from the possibility of being 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 possible cases where part(ii.b) is false, $\mathcal{A}_{paths}$ wins with probability 1, given that part(ii.a) is already false. □

F. Modeling Internal Noise

Here we characterize how the protocols can utilize internal noise, i.e., noise messages generated by a protocol party $\notin \mathcal{S}$, so that we can quantify the bandwidth overhead caused by those noise messages, as well as compare them fairly with noise messages sent by clients. In order to do so, we place the following assumptions on internal noise messages:

1) Since, we do not allow client messages to be dropped, we do not allow internal noises to be dropped.

2) If an internal noise is tagged as a share of message m, that tag can never be changed.

3) Similar to client messages, an internal noise needs to be delivered to the recipient within ℓ rounds from its generation.

4) Internal noise messages must not violate the latency bound of the message that the noise is tagged with. Example: if a node tags a message with A, then the latency of A must be retained, i.e., all messages tagged with A must arrive within ℓ rounds of the round in which the user wanted to send A.

Claim 2 (Internal noise can be replaced with user noise)**.** *For every protocol that uses noise messages originating from internal protocol parties ($\notin \mathcal{S}$) and latency overhead ℓ, there exists a protocol that uses only user generated noise messages (noise messages originating from an user $u \in \mathcal{S}$) and latency overhead $\ell + 1$ with at least equal probability of satisfying Invariant 2.*

Proof sketch. We prove this claim by construction. Given a protocol Π_1 we want to construct a protocol Pi_2 that satisfies the invariant with at least the 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.)

Now two cases can happen:

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 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. □

Ideal Protocol. 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.

Now we construct a protocol Π_{ideal} that intends maximize the probability of satisfying Invariant 2 against $\mathcal{A}_{paths}$, for allowed latency overhead $\hat{\ell}$. The protocol has a number of predefined paths. Those paths are constructed at the beginning of the protocol and do not change throughout the protocol run. Each path is of length $\hat{\ell}$, and consists of $\hat{\ell}$ unique protocol parties, if available. Π_{ideal} has access to an oracle O to decide the number of paths and distribution of protocol parties in each path. We talk about the oracle in more detail later in this section.

Whenever a message (real or noise) is sent to a path it is sent to the protocol party at the position $r \mod \hat{\ell}$ 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 $(r+1) \mod \hat{\ell}$) in the same path. For every message, Π_{ideal} queries the oracle O to decide which path the message should be sent to and the number of rounds the message should remain in the protocol.

If the same user sends multiple messages (real or noise) in a given interval of $\hat{\ell}$ rounds, the objective of the protocol is to send the messages on different paths, so that the total number of protocol parties on the paths of those messages is maximized. Furthermore, shares of the same message should cover as many paths as possible.

Since the protocol has control over the noise messages, it tries to maximize the number of unique users that send the shares for a given message. Additionally, the protocol also tries to maximize the total number of users that send messages in an interval of $\hat{\ell}$ rounds. To achieve the above, Π_{ideal} queries the oracle O for each noise message, and O returns the real message that the noise should be share of.

The oracle O is an overapproximation of different strategies that a protocol can use to optimize paths and noise message. Our oracle knows the user distribution, all past and future messages, the number of compromise 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 the invariant for the given protocol strategy, against the given adversary.

Claim 3 (Ideal protocol is at least 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.*

Proof. 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 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.

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}. □

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.

Now, we derive lower bounds on adversarial advantage for the user distributions as described in Section III-C, so that we can compare our results with the work of Das. et. al. It is worth to repeat here, when all the protocol parties are honest, a protocol without using any secret sharing techniques can perform as good as a protocol that uses secret sharing techniques. Consequently, we will focus on scenarios with compromisation for the remainder of this paper.

VII. Anonymity for synchronized users

In this section we are going to analyze the synchronized user distribution U_B as defined in Section III-C.

A. Lower Bound on Adversarial Advantage

Theorem 5. *For user distribution U_B, no protocol $\Pi \in M$ can provide δ-sender anonymity, for any*
$$\delta < \left(1 - \frac{\beta \mathsf{N}}{\mathsf{N}-1}\right)\left[1 - \frac{(\tau+1)\mathsf{N} - \beta\mathsf{N}\hat{\ell} - \hat{\ell}}{\mathsf{N}} g(\tau) - \frac{\beta\mathsf{N}\hat{\ell} + \hat{\ell} - \tau\mathsf{N}}{\mathsf{N}} g(\tau+1)\right]$$
where $\tau = \lfloor \frac{\beta\mathsf{N}\hat{\ell}+\hat{\ell}}{\mathsf{N}} \rfloor$, $\hat{\ell} = \ell + 1$
and $g(x) = \begin{cases} 1 & c < x\hat{\ell} \\ 1 - \binom{\mathsf{c}}{x\hat{\ell}} / \binom{\mathsf{K}}{x\hat{\ell}} & x\hat{\ell} \leq c. \end{cases}$

Suppose u_0 and u_1 are challenge users, and u_b sends the challenge message. The challenge reaches the recipient at round r. We know from Claim 3 that Π_{ideal} is ideal; thus, we can focus on Π_{ideal} here. By definition of Π_{ideal}, the challenge message can have up to $(\beta\mathsf{N} + 1)$ shares, including the one sent by u_b.

Since we have synchronized user distribution, the best strategy for the oracle O is to have equal number of shares (exactly $(\beta\mathsf{N}+1)$) per real message. Consider any round, there is exactly one real message and $(\beta\mathsf{N} + 1)$ noise messages.

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 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,

$$
\begin{aligned}
&\Pr\left[\text{Invariant 2 is true}\right]\\
&\le \Pr\left[u_{1-b} \text{ sends a share of the challenge message.}\right]\\
&+ Pr[u_{1-b} \text{ does not send a share of the challenge message}\\
&\qquad \wedge\, u_{1-b} \text{ sends a message in the given span of } \hat{\ell} \text{ rounds }]\\
&\quad \times \Pr\left[\text{At least one of the messages visits an honest node}\right]\\
&\le \tfrac{\beta\mathsf{N}}{\mathsf{N}-1} + \left(1-\tfrac{\beta\mathsf{N}}{\mathsf{N}-1}\right) \times \sum_{i=0}^{\infty} \Pr\left[u_{1-b} \text{ sends } i \text{ messages}\right]\\
&\quad \times \Pr\left[\text{at least one of the } i \text{ messages visits an honest node}\right]
\end{aligned}
$$

$$
\begin{aligned}
&\le \tfrac{\beta\mathsf{N}}{\mathsf{N}-1} + \left(1-\tfrac{\beta\mathsf{N}}{\mathsf{N}-1}\right) \tfrac{(\tau+1)\mathsf{N}-\beta\mathsf{N}\hat{\ell}-\hat{\ell}}{\mathsf{N}} \times g(\tau)\\
&\qquad + \left(1-\tfrac{\beta\mathsf{N}}{\mathsf{N}-1}\right) \tfrac{\beta\mathsf{N}\hat{\ell}+\hat{\ell}-\tau\mathsf{N}}{\mathsf{N}} \times g(\tau+1).
\end{aligned}
$$

where $\tau = \lfloor\frac{\beta\mathsf{N}\hat{\ell}+\hat{\ell}}{\mathsf{N}}\rfloor$, 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,

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

Note that, 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 β, $\hat{\ell}$ 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 $\tau = \lfloor\frac{\beta\mathsf{N}\hat{\ell}+\hat{\ell}}{\mathsf{N}}\rfloor$ messages with probability $\frac{(\tau+1)\mathsf{N}-\beta\mathsf{N}\hat{\ell}-\hat{\ell}}{\mathsf{N}}$, and sends $(\tau+1)$ messages with probability $\frac{\beta\mathsf{N}\hat{\ell}+\hat{\ell}-\tau\mathsf{N}}{\mathsf{N}}$.

By Claim 1 whenever Invariant 2 is not true the adversary wins. Hence, we know that the probability that the adversary guesses incorrectly is bounded by: $\Pr\left[0 = \mathcal{A}_{paths}|b=1\right] = \Pr\left[1 = \mathcal{A}_{paths}|b=0\right] \le \frac{1}{2}\Pr\left[\text{Invariant 2 is true}\right]$. Therefore,

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

When $\mathsf{c} < \hat{\ell}$ and $\tau = 0$, we can derive a more precise lower bound on δ than the above one, although the above one is still a valid lower bound on δ. Since $\tau = 0$, there is at most one message sent by u_{1-b} in a span of ℓ 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 $\{r-\mathsf{c}, \dots, r-1\}$. 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 following:

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

B. Impossibility for Strong Anonymity

Theorem 6. *For user distribution U_B with $K, N \in poly(\eta)$, $K > c$, $\hat{\ell} < \mathsf{N}$, $\mathsf{N}-1 > \beta\mathsf{N} \ge 1$, no protocol $\Pi \in$ can achieve strong anonymity if $2\hat{\ell}\beta < 1 - \epsilon(\eta)$, where $\epsilon(\eta) = 1/\eta^d$ for a positive constant d.*
When $\beta\mathsf{N} < 1$, no protocol can achieve strong anonymity, if $2\hat{\ell} < N - \epsilon(\eta)$, instead of $2\hat{\ell}\beta < 1 - \epsilon(\eta)$.
Additionally, when $c \ge \beta\mathsf{N}\hat{\ell}$, strong anonymity can not be achieved if $\beta \le z < \frac{\mathsf{N}-1}{\mathsf{N}} - \epsilon(\eta)$ for any constant z and $\hat{\ell} \in O(1)$.

Proof. We know,

$$
\begin{aligned}
&\Pr\left[\text{Invariant 2 is true}\right]\\
&\le \frac{\beta\mathsf{N}}{\mathsf{N}-1} + \left(1 - \frac{\beta\mathsf{N}}{\mathsf{N}-1}\right) \frac{(\tau+1)\mathsf{N} - \beta\mathsf{N}\hat{\ell} - \hat{\ell}}{\mathsf{N}} \times g(\tau)\\
&\qquad + \left(1 - \frac{\beta\mathsf{N}}{\mathsf{N}-1}\right) \frac{\beta\mathsf{N}\hat{\ell} + \hat{\ell} - \tau\mathsf{N}}{\mathsf{N}} \times g(\tau+1)\\
&= D + (1-D)\left(g(\tau) \times T_1 + g(\tau+1) \times (1-T_1)\right)
\end{aligned}
$$

where $D = \frac{\beta\mathsf{N}}{\mathsf{N}-1}$, $T_1 = \frac{(\tau+1)\mathsf{N}-\beta\mathsf{N}\hat{\ell}-\hat{\ell}}{\mathsf{N}}$.

First, we need to observe that, if $\beta\mathsf{N}\hat{\ell} + \hat{\ell} < \mathsf{N} - \frac{1}{\eta^x}$, strong anonymity can not be achieved. Since, in that case, τ will be zero, and hence, $g(\tau)$ will be zero. Moreover, $T_1 = \frac{\mathsf{N}-\beta\mathsf{N}\hat{\ell}-\hat{\ell}}{\mathsf{N}} > \frac{1}{\mathsf{N}\eta^x}$. Which means, $g(\tau) \times T_1 + g(\tau+1) \times (1 - T_1) < 1 - \frac{1}{\mathsf{N}\eta^x}$ = not overwhelming. Now,

$$
\begin{aligned}
&\beta\mathsf{N}\hat{\ell} + \hat{\ell} < \mathsf{N} - \frac{1}{\eta^x}\\
\Longleftarrow\ & 2\beta\mathsf{N}\hat{\ell} < \mathsf{N} - \frac{1}{\eta^x} \qquad \text{when } \beta\mathsf{N} \ge 1\\
\Longleftrightarrow\ & 2\beta\hat{\ell} < 1 - \frac{1}{\eta^d} \qquad \because \mathsf{N} \in poly(\eta)
\end{aligned}
$$

where d is some constant. Alternatively, when $\beta\mathsf{N} < 1$

$$
\beta\mathsf{N}\hat{\ell} + \hat{\ell} < \mathsf{N} - \frac{1}{\eta^x} \quad \Longleftarrow \quad 2\hat{\ell} < \mathsf{N} - \frac{1}{\eta^x}.
$$

When $c \ge \beta\mathsf{N}\hat{\ell}$, we additionally need both $g(\tau)$ and $g(\tau+1)$ to be overwhelming to achieve strong anonymity, which means both $\tau(\hat{\ell}+1)$ and $(\tau+1)\hat{\ell}$ have to be at least $O(1)$. If both $\hat{\ell}$ and β are in $O(1)$, $\tau = \lfloor\frac{\beta\mathsf{N}\hat{\ell}+\hat{\ell}}{\mathsf{N}}\rfloor = \lfloor\left(\beta\hat{\ell} + \frac{\hat{\ell}}{\mathsf{N}}\right)\rfloor \in O(1)$. Hence, $\tau\hat{\ell}$ is also in $O(1)$. Therefore, $g(\tau)$ and $g(\tau+1)$ are not overwhelming.

□

Similar to Section V, here also we can denote the overhead factor B as $B = \beta\mathsf{N}$. Therefore, we can rewrite the bound on δ as follows —

$$
\delta \ge \left(1 - \tfrac{B}{\mathsf{N}-1}\right)\left[1 - \tfrac{(\tau+1)\mathsf{N}-B\hat{\ell}-\hat{\ell}}{\mathsf{N}} g(\tau) - \tfrac{B\hat{\ell}+\hat{\ell}-\tau\mathsf{N}}{\mathsf{N}} g(\tau+1)\right]
$$

Note that the quantity $g(\tau)$ remains unchanged in this new form (of the bound on δ) as well. Therefore, Theorem 6 can be rewritten as the following:

Corollary 1. *For user distribution U_B with $K, N \in poly(\eta)$, $K > c$, $\hat{\ell} < \mathsf{N}$, $\mathsf{N} - 1 > B \geq 1$, no protocol $\Pi \in$ can achieve strong anonymity if $2\hat{\ell}B < \mathsf{N} - \epsilon(\eta)$, where $\epsilon(\eta) = 1/\eta^d$ for a positive constant d.*
When $B < 1$, no protocol can achieve strong anonymity, if $2\hat{\ell} < N - \epsilon(\eta)$, instead of $2\hat{\ell}B < N - \epsilon(\eta)$.
Additionally, when $c \geq B\hat{\ell}$, strong anonymity can not be achieved if $B \leq N - 1 - \epsilon(\eta)$ and $\hat{\ell} \in O(1)$.

C. Interesting Cases

Now we discuss a selection of cases for different values of $\hat{\ell}$, B, N, c and K.

1) $B \geq \mathsf{N} - 1$: here we can have $\delta = 0$ even for $\hat{\ell} = 0$, for all possible values of c and K. This shows that protocols can achieve anonymity if the bandwidth overhead is sufficient, in the presence of any amount of compromised protocol parties. This is not possible when secret sharing techniques are not used by protocols: even with $\beta = 1$ protocols needed favorable latency overhead, and restriction on c and K

2) $\hat{\ell} = \mathsf{N}$, $B < \mathsf{N} - 1 - \epsilon(\eta)$: In this case, a condition when strong anonymity is possible is if $\mathsf{c} < \tau\hat{\ell}$. However, if $\mathsf{c} > \tau\hat{\ell}$ and $\mathsf{K} - \mathsf{c} = \mathsf{constant}$, strong anonymity is impossible. In a special case, where $\beta\mathsf{N} < 1$, $\tau \approx \lfloor 1 + \beta\mathsf{N} \rfloor = 1$. In that case, we need $\mathsf{c} < \hat{\ell}$ or $(\mathsf{K} - \mathsf{c}) \in \omega(1)$. Which means, $\hat{\ell}$ by itself can not provide strong anonymity, unless complemented by β or enough number of honest parties.

3) $\mathsf{K} = poly(\eta), \hat{\ell} = \mathsf{K}, \tau = 1, \mathsf{c} = \mathsf{K} - 1$: In this case, strong anonymity is possible — which is not possible for similar conditions if protocols can not use secret sharing techniques. Since, the protocols can use secret sharing technique, the adversary needs compromise more protocol parties to break strong anonymity. Which shows that secret sharing can provide resistance against compromisation.

4) $\mathsf{K} = poly(\eta), \tau\hat{\ell} \geq \mathsf{c}, \mathsf{K} - \mathsf{c} = constant$: This is a slightly generic case of our previous example. Here as well, since $\tau\hat{\ell} \geq \mathsf{c}$, the factors $g(\tau)$ and $g(\tau + 1)$ will evaluate to 1. Therefore, strong anonymity is possible. Actually, the relationship between K and c is immaterial here, as long as $\tau\hat{\ell} \geq \mathsf{c}$. If protocols do not use secret sharing techniques, we have seen that the relation between K and c is very important for anonymity.

5) $\mathsf{K} = poly(\eta), \tau\hat{\ell} < \mathsf{c}, \mathsf{K} - \mathsf{c} = constant$: This is similar to the previous case, except now we have $\tau\hat{\ell} < \mathsf{c}$. Since $\mathsf{K} - \mathsf{c} = constant$, $g(\tau)$ or $g(\tau + 1)$ can never be overwhelmingly 1. Hence, δ can never be negligible for any $\beta N < N - 1 - \epsilon(\eta)$. Therefore, strong anonymity is impossible without a trivially large bandwidth overhead. With increasing amount of compromisation, the latency and bandwidth overhead need to increase accordingly to compensate the amount of compromisation.

6) ${}^{\mathsf{K}}/_{\mathsf{c}} = constant$: This scenario is especially interesting, because the impossibility/possibility of anonymity depends on $\hat{\ell}$ and B.

(a) Needless to mention, when β is 1, the chance of strong anonymity is obvious. We shall consider the more interesting case where $\beta < 1$.
(b) If $\hat{\ell} \in O(1)$ and $\mathsf{c} > \hat{\ell}$, it is impossible for any protocol to achieve strong anonymity.
(c) If we are happy with a weaker form of anonymity, choosing $\hat{\ell} \in \Theta(log(\eta))$ might be a good tradeoff. The adversarial advantage will be bounded by ${}^{1}/_{poly(\eta)}$ instead of $neg(\eta)$, but only when ${}^{\mathsf{K}}/_{\mathsf{c}}$ is large enough. For instance, if $\hat{\ell} = log(\eta)$ and τ is not polynomially large, any ${}^{\mathsf{K}}/_{\mathsf{c}} < 2$ will result in $\delta > \Theta({}^{1}/_{\eta})$, given $\left(1 - \frac{\beta\mathsf{N}}{(\mathsf{N}-1)}\right)$ is not polynomially small.
(d) However, for $\hat{\ell} = log(\eta)$, if we pick ${}^{\mathsf{K}}/_{\mathsf{c}} > 4$, there is a possibility that the adversarial advantage will be bounded by $O({}^{1}/_{\eta^2})$, for any $\tau \in O(1)$ and any given value of $\left(1 - \frac{\beta\mathsf{N}}{(\mathsf{N}-1)}\right)$.
(e) If we pick $\hat{\ell} = log^b(\eta)$, and ${}^{\mathsf{K}}/_{\mathsf{c}} > 2$, there is a possibility that the adversarial advantage will be bounded by $O({}^{1}/_{\eta^b})$, for any $\tau \in O(1)$ and any given value of $\left(1 - \frac{\beta\mathsf{N}}{(\mathsf{N}-1)}\right)$.

VIII. Anonymity for unsynchronized users

Now, we are going to analyze the unsynchronized user distribution U_P as defined in Section III-C. To briefly reiterate, in this user distribution each user act independent of other users. Each user tosses a coin with success probability $p \in (0, 1]$ in every round to decide whether to send a message or not, independent of other rounds and any other user. Here, we assume that the bandwidth overhead β to be a part of p. If the user wants to send messages in each round with probability p' ($p' < p$), then bandwidth overhead is $\beta = p - p'$ per user in each round, or $B = \frac{p-p'}{p'}$ noise messages per real message.

A. Lower Bound on Adversarial Advantage

Theorem 7. *For user distribution U_P, no protocol $\Pi \in M$ can provide δ-sender anonymity, for any*

$$\delta < \begin{cases} \left(1 - \frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\left[1 - g(Z) \times \left(1 - (1-p)^{\hat{\ell}}\right)\right], & \mathsf{c} \geq \hat{\ell} \\ \left(1 - \frac{B_{\mathit{eff}}}{\mathsf{N}-1}\right) \times (1-p)^{\hat{\ell}-\mathsf{c}} \Bigg[1 - \left(1 - (1-p)^{\mathsf{c}}\right) & \\ \times\left(\Pr\left[W \geq 1\right] + \Pr\left[W = 0\right]\left[1 - 1/\binom{\mathsf{K}}{\mathsf{c}}\right]\right)\Bigg] & \mathsf{c} < \hat{\ell} \end{cases}$$

where $B_{\text{eff}} = min(B, \hat{\ell}p - 1)$, $Z = min(\hat{\ell}, 2B_{\text{eff}} + 1)$,
W is a random variable denoting the number of additional shares for the challenge message, and

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

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.

Let us define $B_{\text{eff}} = min(B, \hat{\ell}p - 1)$. If we look at Invariant 2, the shares sent at rounds $\{(r - \hat{\ell}), \ldots, (r - 1)\}$

can contribute anonymity, but not the ones sent before round $(r-\hat{\ell})$. That is why the number of shares influencing anonymity is limited by $\hat{\ell}p-1$, even though B is really high.

Additionally, if W is a random variable denoting the number of shares of the challenge message excluding the share sent by the challenge user, using Chernoff bound we know, $\Pr[W \geq 2\mathbb{E}[W]] = \Pr[W \geq 2B] \leq \exp\left(-2\frac{\mathbb{E}[W]^2}{\mathsf{N}^2}\mathsf{N}\right)$, where $W = \sum_{i=1}^{N} W_i$ with W_i denoting the number of shares sent by the i-th user. Note that W is a random variable, where $W = min\left(\hat{\ell}(X-X'), (X-X')/X'\right)$. Here X and X' follow $Binom(\hat{\ell},p)$ and $Binom(\hat{\ell},p')$ respectively. $\Pr[W \geq 2\mathbb{E}[W]]$ is negligible in N, and hence negligible in η, so long as $\mathbb{E}[W]$ is not negligible. Let us denote, $Z = min(\hat{\ell}, 2B_{\text{eff}}+1)$.

With the above in our hand, for $\mathsf{c} \geq \hat{\ell}$ we can derive the following:

$$\begin{aligned}
&\Pr[\text{Invariant 2 is true}]\\
&\leq \Pr[u_{1-b} \text{ sends a share of the challenge message.}]\\
&\quad + Pr[u_{1-b} \text{ does not send a share of the challenge message}\\
&\qquad \wedge\, u_{1-b} \text{ sends a message in the given span of round } \hat{\ell}]\\
&\qquad\quad \times Pr[\text{Some share of the challenge message visits honest node}\\
&\qquad\qquad \text{and some message from } u_{1-b} \text{ visits honest node}]\\
&\leq \frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1-\frac{B_{\text{eff}}}{\mathsf{N}-1}\right) Pr\left[\text{At least one honest node in Z paths}\right]\\
&\quad \times \Pr\left[u_{1-b} \text{ sends at least one message in}\{(r-\hat{\ell}),\ldots,(r-1)\}\right]\\
&\leq \frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1-\frac{B_{\text{eff}}}{\mathsf{N}-1}\right) \times g(Z) \times \left(1-(1-p)^{\hat{\ell}}\right)
\end{aligned}$$

By Claim 1, the adversary wins whenever Invariant 2 is not true. Hence, the advantage of the adversary is bounded by: $\delta \geq 1 - \Pr[\text{Invariant 2 is true}] \geq \left(1-\frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\left[1-g(Z)\times\left(1-(1-p)^{\hat{\ell}}\right)\right]$.

When $\mathsf{c} < \hat{\ell}$, we can derive the following,

$$\begin{aligned}
&\Pr[\text{Invariant 2 is true}]\\
&\leq \Pr[u_{1-b} \text{ sends a share of the challenge message.}]\\
&\quad + Pr[u_{1-b} \text{ does not send a share of the challenge message}]\\
&\quad\ \wedge \Big(Pr[u_{1-b} \text{ sends a message in } \{r-\hat{\ell}, r-\mathsf{c}-1\}]\\
&\quad\ + Pr[u_{1-b} \text{ does not send a message in } \{r-\hat{\ell}, r-\mathsf{c}-1\}]\\
&\quad\ \times \Big(Pr[u_{1-b} \text{ sends more than one message in } \{r-\mathsf{c}, r-1\}]\\
&\quad\ + Pr[u_{1-b} \text{ sends only one message in } \{r-\mathsf{c}, r-1\}]\\
&\quad\ \times Pr[\text{Some share of the challenge message visits honest node}\\
&\qquad\quad \text{and some message from } u_{1-b} \text{ visits honest node}]\Big)\Big)\\
&\leq \frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1-\frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\left[\left(1-(1-p)^{\hat{\ell}-\mathsf{c}}\right) + (1-p)^{\hat{\ell}-\mathsf{c}}\right.\\
&\quad \times \Big(\Pr[W \geq 1 \wedge X^{u_{1-b}}(c) \geq 2]\\
&\quad \left. + \Pr[W = 0 \wedge X^{u_{1-b}}(c) \geq 1] \times \left[1 - 1/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\Big)\right]\\
&\leq \frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1-\frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\left[\left(1-(1-p)^{\hat{\ell}-\mathsf{c}}\right) + (1-p)^{\hat{\ell}-\mathsf{c}}\right.\\
&\times \Pr[X^{u_{1-b}}(c) \geq 1]\left.\left(\Pr[W \geq 1] + \Pr[W = 0]\left[1-1/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\right)\right]\\
&\leq \frac{B_{\text{eff}}}{\mathsf{N}-1} + \left(1-\frac{B_{\text{eff}}}{\mathsf{N}-1}\right)\left[\left(1-(1-p)^{\hat{\ell}-\mathsf{c}}\right) + (1-p)^{\hat{\ell}-\mathsf{c}}\right.\\
&\quad \times (1-(1-p)^{\mathsf{c}})\left.\left(\Pr[W \geq 1] + \Pr[W = 0]\left[1-1/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\right)\right]
\end{aligned}$$

Thus,

$$\begin{aligned}
&\delta \geq 1 - \Pr[\text{Invariant 2 is true}]\\
&\geq \left(1-\tfrac{B_{\text{eff}}}{\mathsf{N}-1}\right) \times (1-p)^{\hat{\ell}-\mathsf{c}}\left[1-\left(1-(1-p)^{\mathsf{c}}\right)\right.\\
&\quad \times \left.\left(\Pr[W \geq 1] + \Pr[W = 0]\left[1-1/\tbinom{\mathsf{K}}{\mathsf{c}}\right]\right)\right].
\end{aligned}$$

B. Impossibility for Strong Anonymity

We can observe that, to achieve strong anonymity, we need $B_{\text{eff}} > (\mathsf{N}-1) - neg(\eta)$ or, $c \leq Z\hat{\ell}$. $B_{\text{eff}} > (\mathsf{N}-1) - neg(\eta)$ is a trivial condition, which implies very high bandwidth overhead.

However, $c \leq Z\hat{\ell}$ is not very trivial. We know that, $Z = min(\hat{\ell}, 2B_{\text{eff}}+1)$, where $(2B_{\text{eff}}+1)$ represents an upper bound on the number of shares per real message in a given interval of $\hat{\ell}$ rounds.

Theorem 8. *For user distribution U_P, no protocol $\Pi \in M$ can achieve strong anonymity if $B < (\mathsf{N}-1) - \epsilon(\eta)$ and $p\hat{\ell} < 1 - \epsilon(\eta)$ and $\mathsf{c} > \hat{\ell}^2$ and $\hat{\ell}^2 \in O(1)$.*

Proof sketch. If $B < (\mathsf{N}-1) - \epsilon(\eta)$, $\frac{B}{\mathsf{N}-1}$ will be less than $1 - neg(\eta)$. Hence $H = \left[1-g(Z)\times\left(1-(1-p)^{\hat{\ell}}\right)\right]$ has to be negligible to achieve strong anonymity.

When $p\hat{\ell} < 1 - \epsilon(\eta)$, $\left(1-(1-p)^{\hat{\ell}}\right)$ can never be overwhelming, and consequently, H can never be negligible.

Even when $\left(1-(1-p)^{\hat{\ell}}\right)$ is overwhelming, $g(Z)$ has to be overwhelming as well to achieve strong anonymity, which implies $\left[\binom{\mathsf{c}}{Z\hat{\ell}}/\binom{\mathsf{K}}{Z\hat{\ell}}\right]$ has to be negligible (since $\mathsf{c} \geq \hat{\ell}^2 \implies$

$\mathsf{c} \geq Z\hat{\ell}$), to achieve strong anonymity. $\binom{\mathsf{c}}{Z\hat{\ell}}/\binom{\mathsf{K}}{Z\hat{\ell}}$ can never be negligible if $\hat{\ell}^2 \in O(1)$, since $Z \in O(\hat{\ell})$. □

In a similar way we can prove the corollaries as well.

Corollary 2. *For user distribution U_P, no protocol $\Pi \in M$ can achieve strong anonymity if $B \in O(1)$ and $c > (2B+1)\hat{\ell}$ and $\hat{\ell} \in O(1)$.*

Corollary 3. *For user distribution U_P and $B = 0$, no protocol $\Pi \in M$ can achieve strong anonymity if and $c > \hat{\ell}$ and $\hat{\ell} \in O(1)$.*

Theorem 9. *For user distribution U_P, $p < 1 - \epsilon(\eta)$, $\frac{\mathsf{c}}{\mathsf{K}} = const$, no protocol $\Pi \in M$ can achieve strong anonymity if $B_{\text{eff}} < (\mathsf{N}-1) - \epsilon(\eta)$ and $c > \hat{\ell}^2$ and $\hat{\ell}^2 \in O(log(\eta))$, where $\epsilon(\eta) = 1/\eta^x$ for a positive constant x.*

Proof. Note that, we can rewrite,
$$\Pr\left[\text{Invariant 2 is true}\right] \leq \frac{B}{(\mathsf{N}-1)} + \left(1 - \frac{B}{(\mathsf{N}-1)}\right) \times \sum_{d=0}^{\hat{\ell}} \left[1 - \binom{\mathsf{c}}{d\ell'}/\binom{\mathsf{K}}{d\ell'}\right] \left[\binom{\hat{\ell}}{d} p^d (1-p)^{\hat{\ell}-d}\right]$$

If $B < (\mathsf{N}-1) - \epsilon(\eta)$, $\frac{\beta}{\mathsf{N}-1}$ will be less than $1 - neg(\eta)$. In that case, since ℓ' is always upper bounded by $\hat{\ell}$, $D = \sum_{d=0}^{\hat{\ell}} \left[1 - \binom{\mathsf{c}}{d\hat{\ell}}/\binom{\mathsf{K}}{d\hat{\ell}}\right] \left[\binom{\hat{\ell}}{d} p^d (1-p)^{\hat{\ell}-d}\right]$ has to be overwhelming to achieve strong anonymity. We know, $\sum_{d=0}^{\hat{\ell}} \left[\binom{\hat{\ell}}{d} p^d (1-p)^{\hat{\ell}-d}\right] = 1$. Therefore, for D to become overwhelming, we need $\left[1 - \binom{\mathsf{c}}{d\hat{\ell}}/\binom{\mathsf{K}}{d\hat{\ell}}\right]$ to be overwhelming for each d, whenever $\left[\binom{\hat{\ell}}{d} p^d (1-p)^{\hat{\ell}-d}\right]$ is non-negligible. Note that, since both $\hat{\ell}$ and $d \leq \hat{\ell}$ are in $poly(\eta)$, $d\hat{\ell}$ is in $O(\hat{\ell}^2)$.

We know, $\frac{\mathsf{c}}{\mathsf{K}} = const = \frac{1}{y}$. Therefore, we can say,

$$\frac{\mathsf{c} - d\hat{\ell}}{\mathsf{K} - d\hat{\ell}} > \frac{1}{y} \iff \left(\frac{\mathsf{c} - d\hat{\ell}}{\mathsf{K} - d\hat{\ell}}\right)^{d\hat{\ell}} > \left(\frac{1}{y}\right)^{d\hat{\ell}}$$
$$\implies \frac{\mathsf{c}(\mathsf{c}-1)\dots(\mathsf{c}-d\hat{\ell})}{\mathsf{K}(\mathsf{K}-1)\dots(\mathsf{K}-d\hat{\ell})} > \left(\frac{\mathsf{c} - d\hat{\ell}}{\mathsf{K} - d\hat{\ell}}\right)^{d\hat{\ell}} > \left(\frac{1}{y}\right)^{d\hat{\ell}}$$
$$\iff \frac{\binom{\mathsf{c}}{d\hat{\ell}}}{\binom{\mathsf{K}}{d\hat{\ell}}} > \left(\frac{1}{y}\right)^{d\hat{\ell}}.$$

$\left(\frac{1}{y}\right)^{d\hat{\ell}}$ can never be negligible for when $d\hat{\ell}$ is in $O(log(\eta))$ and $c > d\hat{\ell}$ — which is bound to happen if $\hat{\ell}^2 \in O(log(\eta))$ and $c > \hat{\ell}^2$. □

Theorem 10. *For user distribution U_P, for $B < (\mathsf{N}-1) - \epsilon(\eta)$ and $p(\hat{\ell} - \mathsf{c}) < 1 - \epsilon(\eta)$, no protocol $\Pi \in M$ can achieve strong anonymity if $p\mathsf{c} < 1 - \epsilon(\eta)$ OR $\mathsf{c} \in O(1)$.*

Proof Sketch. 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 $p\mathsf{c} < 1 - \epsilon(\eta)$ or $\mathsf{c} \in O(1)$. □

C. Interesting Cases

Similar to Section VII, now we discuss a few examples for different values of $\hat{\ell}$, β, N, c and K.

1) $B \geq N$: here we can have $\delta = 0$ even for $\hat{\ell} = 0$, for all possible values of c and K. This can happen because for each real message there are at least N noise messages, and each of them can be a share of the real message. This shows that protocols can achieve anonymity, given enough bandwidth overhead, in the presence of any amount of compromised protocol parties (except u_b and u_{1-b}). This was not possible when secret sharing techniques are not used by protocols, even with $\beta = 1$ protocols needed favorable latency overhead, and restriction on c.

2) $\hat{\ell} = \mathsf{N}$, $B < \mathsf{N} - 1$: strong anonymity is possible if $\mathsf{c} < Z\hat{\ell}$. Which means, ℓ by itself can not provide strong anonymity, unless complemented by β or enough number of honest parties.

3) $\mathsf{K} = poly(\eta), \hat{\ell} = \mathsf{K}, p = {}^{1}/_{\mathsf{K}}, \mathsf{c} = \mathsf{K} - 1$: In this case, strong anonymity is possible — which is not possible for similar conditions if protocols can not use secret sharing techniques. Which again shows, the adversary needs to compromise more protocol parties to break strong anonymity, if the protocols can use secret sharing technique.

4) $\mathsf{K} = poly(\eta), 2p\ell^2 \geq \mathsf{c}, \mathsf{K} - \mathsf{c} = constant$: This is a slightly generic case of our previous example. Since $2p\ell^2 \geq \mathsf{c}$, the factor $g(2p\ell)$ will evaluate to 1. Therefore, strong anonymity is possible. Actually, the relationship between K and c is immaterial here, as long as $\tau\ell \geq \mathsf{c}$. If protocols do not use secret sharing techniques, we have seen that the relation between K and c is very important for anonymity.

5) $\mathsf{K} = poly(\eta), 2p\hat{\ell}^2 < \mathsf{c}, \mathsf{K} - \mathsf{c} = constant$: This is similar to the previous case, except now we have $2p\hat{\ell}^2 < \mathsf{c}$. Since $\mathsf{K} - \mathsf{c} = constant$, $g(2p\hat{\ell})$ can never be overwhelmingly 1. Hence, δ can never be negligible for any $B < N - 1 - {}^{1}/_{poly(\eta)}$. Therefore, strong anonymity is impossible without a trivially large bandwidth overhead. With increasing amount of compromisation, the latency and bandwidth overhead need to increase accordingly to compensate the amount of compromisation.

6) ${}^{\mathsf{K}}/_{\mathsf{c}} = constant$: This scenario is especially interesting, because the impossibility/possibility of anonymity depends on $\hat{\ell}$ and B.
(a) Needless to mention, when B is N, the chance of strong anonymity is obvious. We shall consider the more interesting case where $B < N$.
(b) If $\hat{\ell} \in O(\eta)$ and $\mathsf{c} > \hat{\ell}$, it is impossible for any protocol to achieve strong anonymity.
(c) If we are happy with a weaker form of anonymity, choosing $\hat{\ell} \in \Theta(log(\eta))$ might be a good tradeoff. The adversarial advantage will be bounded by ${}^{1}/_{poly(\eta)}$ instead of $neg(\eta)$, but only when ${}^{\mathsf{K}}/_{\mathsf{c}}$ is large enough. For instance, if $\hat{\ell} = log(\eta)$, any ${}^{\mathsf{K}}/_{\mathsf{c}} < 2$ will result in $\delta > \Theta({}^{1}/_{\eta})$, given $\left(1 - \frac{B}{(\mathsf{N}-1)}\right)$ is not polynomially small.

(d) However, for $\hat{\ell} = log(\eta)$, if we pick ${}^{\mathsf{K}}\!/_{\mathsf{c}} > 4$, there is a possibility that the adversarial advantage will be bounded by $O({}^{1}\!/_{\eta^2})$, for any given value of $\left(1 - \frac{B}{(\mathsf{N}-1)}\right)$.

(e) If we pick $\hat{\ell} = log^b(\eta)$, and ${}^{\mathsf{K}}\!/_{\mathsf{c}} > 2$, there is a possibility that the adversarial advantage will be bounded by $O({}^{1}\!/_{\eta^b})$, for any given value of $\left(1 - \frac{B}{(\mathsf{N}-1)}\right)$.

D. Modified User Distribution

For secret-sharing based ACNs a different noise distribution is more beneficial for the protocols. In the spirit of over-approximating the protocol's capabilities, we assume that for each real message that is sent, up to B noise messages are generated from other clients, in contrast to previously sending a noise message with probability $p - p'$. Since messages can not be sent in parts, we assume that B is an integer $\in \{0, \ldots, \mathsf{N} - 1\}$. Let us call this user distribution U_m.

The ideal protocol would in this setting try to maximize the number of users sending B noise messages; additionally it will maximize the total number of users sending messages in a given span of $\hat{\ell}$ rounds.

In this case, we can define the expected number of messages sent by user i in a round as $p = (p'\mathsf{N} + p'B\mathsf{N})/\mathsf{N} = p' + p'B$. Therefore, in a span of $\hat{\ell}$ rounds, the expected number of messages sent by u_{1-b} is $\hat{\ell}p$.

Suppose, $X^{(i)}$ is a random variable denoting the number of real messages sent by user i in a span of $\hat{\ell}$ rounds. Using Chernoff bound on $X = \sum_{i=1}^{\mathsf{N}} X^{(i)}$, we can prove that X is bounded by $2\mathbb{E}[X]$ with overwhelming probability. If we denote the total number of messages sent by u_{1-b} is a span of $\hat{\ell}$ rounds with a random variable Y, then $Y = X(1 + B)/\mathsf{N}$. Therefore, Y will be bounded by $2\mathbb{E}[Y] = 2\hat{\ell}p$ with overwhelming probability.

Then, we can derive the following bounds on adversarial advantage in a way very analogous to Section VIII-A.
when $\mathsf{c} < \hat{\ell}$, $B = 0$:
$\delta \geq \left(1 - \frac{B}{\mathsf{N}-1}\right)(1-p)^{\hat{\ell}-\mathsf{c}}\left[1 - \left(1 - (1-p)^{\mathsf{c}}\right)\left[1 - 1/\binom{\mathsf{K}}{\mathsf{c}}\right]\right]$
when $\mathsf{c} < \hat{\ell}$, $B \geq 1$:
$\delta \geq \left(1 - \frac{B}{\mathsf{N}-1}\right)(1-p)^{\hat{\ell}-\mathsf{c}}(1-p)^{\mathsf{c}}$.
When $\mathsf{c} \geq \hat{\ell} : \delta \geq \left(1 - \frac{B}{\mathsf{N}-1}\right)\left[1 - g(\hat{Z}) \times \left(1 - (1-p)^{\hat{\ell}}\right)\right]$, where $\hat{Z} = min(\hat{\ell}, 2\ell p, B)$. Note that, in the lower bound of δ, we are using $Bprob$ and not $Bprob'$.

Analogously we can derive the following impossibility theorems which are very similar to those in Section VIII-B.

Theorem 11. *For user distribution U_m, no protocol $\Pi \in M$ can achieve strong anonymity if $B < (\mathsf{N}-1)$ and $p\hat{\ell} < 1-\epsilon(\eta)$ and $\mathsf{c} > \hat{\ell}^2$ and $\hat{\ell}^2 \in O(1)$.*

Theorem 12. *For user distribution U_m, $p < 1 - \epsilon(\eta)$, $\frac{\mathsf{c}}{\mathsf{K}} =$ const, no protocol $\Pi \in M$ can achieve strong anonymity if $B < (\mathsf{N} - 1)$ and $c > \hat{\ell}^2$ and $\hat{\ell}^2 \in O(log(\eta))$, where $\epsilon(\eta) = 1/\eta^x$ for a positive constant x.*

Theorem 13. *For user distribution U_m, for $B < (\mathsf{N} - 1)$ and $p(\hat{\ell} - \mathsf{c}) < 1 - \epsilon(\eta)$, no protocol $\Pi \in M$ can achieve strong anonymity if $p\mathsf{c} < 1 - \epsilon(\eta)$ or $\mathsf{c} \in O(1)$.*

The proofs and analysis of the above impossibility conditions are very similar to that in Section VIII-B, the only differences are that we use B instead of B_{eff} in this setting, and we use newly defined $\hat{Z}$ instead of Z. Hence, we refer the readers to Section VIII-B for a detailed account on those.

E. Improved bound for classical protocols

As a byproduct of our new lower bound for δ for secret sharing protocols, we also achieve a new and improved bound for classical protocol for user distributions U_P and U_m. Instead of providing a tedious and long derivation, we provide a short intuition here about how the bounds are derived for U_P. In Invariant 2, there can be multiple paths for the messages from u_{1-b}, also for the shares of the challenge message. However, if we reduce Invariant 2 to Invariant 1, there can be only one path for the challenge message. Additionally, the protocol does not gain anything for anonymity from the shares of the challenge message, because there are no shares.

Therefore, the adversarial advantage δ for $\mathsf{c} < \ell$ in case of U_P can be written as:

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

Similar for $\mathsf{c} \geq \ell$, $\delta \geq 1 - \left(1 - (1-p)^{\ell}\right)\left[1 - \binom{\mathsf{c}}{\ell}/\binom{\mathsf{K}}{\ell}\right]$.

IX. Analysis of Results

A. Impossibility Results

From our impossibility theorems in Sections VII and VIII, we observe that strong anonymity requires a combination of latency and bandwidth overhead - which is very similar to the observations from Das. et. al. [13]. However, secret-sharing based protocols improves on the cost to achieve anonymity. As a special example, with the aid of secret sharing techniques, strong anonymity can be achieved for $\beta > 1$ (or $p > 1$) – which is not possible for protocols that do not use any secret sharing. In Table II we compare the impossibility conditions for anonymity for protocols with secret sharing against classical (mix-net-type) protocols. Whenever the conditions in a line in Table II are met, strong anonymity is impossible, e.g., for synchronized user distribution for mixnets, if $\mathsf{c} > \ell$ and either of $\ell \in O(1)$ or $2\ell B < \mathsf{N} - \epsilon(\eta)$ is true, strong anonymity is impossible.

For secret sharing based protocols, we additionally observe that bandwidth overhead can provide resistance against compromised parties. In the previous case (non-secret-sharing scenario), bandwidth only compensated for latency.

TABLE II

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 $\ell < \mathsf{N}$ and $(\mathsf{N} - 1) - \epsilon(\eta) \geq B \geq 1$ and $\epsilon(\eta) = 1/\eta^d$ for a positive constant d. We compare $\ell = x$ of mix-net type protocols with $\hat{\ell} = x$ of protocols with secret sharing; and we denote the case with $\ell = x$ (See Footnote 4) in the leftmost column. All other columns shows the impossibility conditions for anonymity for the combination of user distribution and protocol class. Where two rows have overlapping cases (leftmost column), if either of the conditions are true, strong anonymity is impossible.

Cases	U_B Classical	U_P Classical	U_B, with secret sharing	U_P, with secret sharing
$\mathsf{c} \geq 0$	$2\ell B < \mathsf{N} - \epsilon(\eta)$	$2\ell p < 1 - \epsilon(\eta)$	$2\hat{\ell} B < \mathsf{N} - \epsilon(\eta)$	$p\hat{\ell} < 1 - \epsilon(\eta)$
$B < 1$	$2\ell B < \mathsf{N} - \epsilon(\eta)$	$2\ell p < 1 - \epsilon(\eta)$	$2\hat{\ell} < N - \epsilon(\eta)$	$p\hat{\ell} < 1 - \epsilon(\eta)$
$0 < \mathsf{c} \leq \ell$	$2(\ell - \mathsf{c})B < \mathsf{N} - \epsilon(\eta)$	$2(\ell - \mathsf{c})p < 1 - \epsilon(\eta)$	$2(\hat{\ell} - \mathsf{c})B < \mathsf{N} - \epsilon(\eta)$	$p(\hat{\ell} - \mathsf{c}) < 1 - \epsilon(\eta)$
$\ell < \mathsf{c} \leq B\ell$	$\ell \in O(1)$	$\ell \in O(1)$	$2(\hat{\ell} - \mathsf{c})B < \mathsf{N} - \epsilon(\eta)$	$p(\hat{\ell} - \mathsf{c}) < 1 - \epsilon(\eta)$
$B\ell < \mathsf{c} \leq \ell^2$	$\ell \in O(1)$	$\ell \in O(1)$	$\hat{\ell} \in O(1)$	$p(\hat{\ell} - \mathsf{c}) < 1 - \epsilon(\eta)$
$\mathsf{c} > \ell^2$	$\ell \in O(1)$	$\ell \in O(1)$	$\hat{\ell} \in O(1)$	$\hat{\ell} \in O(1)$
$\mathsf{K}/\mathsf{c} \in O(1)$	$\ell \in log(\eta)$	$\ell \in log(\eta)$	$\hat{\ell}^2 \in log(\eta)$	$\hat{\ell}^2 \in log(\eta)$

B. Interesting Cases

We next discuss a series of interesting cases for anonymous communication protocols with secret sharing. For comparison purpose, we take cases which are canonical to the cases in **??**. When we compare the results from classical scenarios with the results of secret-sharing scenarios, we compare $\ell = x$ of classical with $\hat{\ell} = x$ of secret-sharing scenario to induce fairness.[4]. We refer to Table III for a concise overview.

X. Implications

Our novel necessary constraints for the core of secret-sharing based ACNs 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 contraints. This section discusses some of these ACNs, in particular whether they utilize advantages of secret-sharing based ACNs that lead to the differences in this work's necessary constraints for mix-nets and secret-sharing based ACNs.

Chaum started a line of work on so-called DC-nets [7], [21], [23], [24]. DC-nets provide an anonymous broadcast channel. In DC-nets, in every round, each party broadcasts either a noise or real message to all clients. With our formalism that means $B = N$, i.e., every real message incurs an overhead of N messages. In DC-nets, pair-wise shared keys among the clients are used to create noise messages that cancel out in an XOR combination of all messages. To avoid collisions in the broadcast channel, i.e., that two real messages disturb each other, usually some cryptographic scheduling protocol is run such that in each round only one party is scheduled to send a message but each party only knows whether it is its own turn but not which other party's turn it is. The communication complexity of this protocol is very low. The latency is $1 = \ell$ round (ignoring the scheduling protocol). With $B = N$ and $\ell = 1$, DC-nets satisfy our novel necessary constraints for secret-sharing based ACNs from Theorem 9 and escape our impossibility results.

The bandwidth and communication overhead of DC-nets are tremendous, as every client sends a message to every other client in each round. The ACN Dissent-AT [7] (the AnyTrust-variant of Dissent) improves among other features the communication overhead of DC-nets by relying on K computation servers (which constitute the K protocol parties in terms of our formalism), assuming that at least one of these servers is honest. Using secret-sharing and assuming a shared secret with each of the K servers, Dissent-AT then achieves that each client only has to send each real or noise message to one of the K servers and not to all other clients. Afterwards, these K servers broadcast these shares to each other. 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). For our formalism the bandwidth overhead is $B = N$ just as for DC-nets. Hence, Dissent-AT also satisfies our necessary constraints for secret-sharing based ACNs from Theorem 9 and escapes our impossibility results.

Dicemix [10] is a peer-to-peer AC protocol that is based on the DC-net approach. While Dicemix includes a self-healing mechanism that leads to $4 + 2f$ communication rounds for one message if f peers are malicious, this mechanism does not kick in if all peers are honest, leading to only 4 communication rounds, resulting in $\ell \in \theta(1)$. As every party sends a message in every round, $\beta \in \theta(\mathsf{N}/\mathsf{N})$. For the same reasons as DC-nets, Dicemix escapes our impossibility results.

There is a recent line of work [8], [22], [25] that uses secret-sharing (e.g., inside modified private information retrieval protocols) to achieve strong anonymity in the presence of compromised parties. These protocols, however, fail to achieve the property that we assume: even for the recipients of the message the packet sent by the real sender and the dummy messages are indistinguishable.

In conclusion, none of the ACNs of which we are aware utilize the mix of multi-hop layered encryption feature, as used in mix-nets, with secret-sharing like features that render the real sender's packet indistinguishable form a noise message

[4] When we allow latency to be $\ell + 1$ for secret-sharing scenarios 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 classical protocols with latency ℓ. Moreover, when $\ell = 0$, there is no intermediate party, so there is no internal noise.

TABLE III
Interesting cases for Anonymous Communication (with secret sharing techniques), 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, and $\epsilon(\eta) = 1/\eta^d$ for a positive constant d. In the rows labeled with "Ano." we show whether strong anonymity might be possible ("strong"), whether quadratic anonymity ($\delta < \frac{1}{\eta^2}$) might be possible ("quad") or whether neither of them are possible ("none"). In each row labeled with "Add. req." we describe which additional requirements (for the respective degree of anonymity) we show. We compare $\ell = x$ of classical with $\hat{\ell} = x$ of secret-sharing scenario; and we denote the case with $\ell = x$ (See Footnote 4) for ease of notation.

	U_B Classical		U_B Secret sharing		U_P Classical		U_P Secret sharing	
Cases	Ano.	Add. req.	Ano.	Add. req.	Ano.	Add. req.	Ano.	Add. req.
$B = \mathsf{N}, \ell = 0^4, \mathsf{c} = 0$	none		strong		none		strong	$p\ell$
$B = \mathsf{N}, \ell = 0, \mathsf{c} = \mathsf{K}$	none		strong		none		strong	
$B = \mathsf{N}, \ell = 1, \mathsf{c} = 0$	strong		strong		strong	$p'\ell \geq 1$	strong	$p'\ell \geq 1$
$B = \mathsf{N}, \ell = 1, \mathsf{c} = \mathsf{K}/2$	quad	$\mathsf{K} \in \Omega(\eta)$	strong		strong	$\mathsf{K} \geq \eta$, $p'\ell \geq 1$	strong	$p'\ell \geq 1$
$B = \mathsf{N}, \ell = 1, \mathsf{c} = \mathsf{K} - 1$	quad	$\mathsf{K} \in \Omega(\eta)$	strong		quad	$\mathsf{K} \geq \eta^2$	strong	
$B = \mathsf{N}, \ell = 1, \mathsf{c} = \mathsf{K}$	none		strong	$p'\ell \geq 1$	none		strong	$p'\ell \geq 1$
$\ell = \mathsf{K}, B\ell = N, \mathsf{c} = \mathsf{K}/2, K = \frac{\mathsf{N}}{2}$	quad	$\mathsf{N} \in \Theta(\eta)$	quad	$\mathsf{N} \in \Theta(\eta)$	strong	$p > \frac{2\eta}{\mathsf{K}}$	strong	$p < \frac{2\eta}{\mathsf{K}}$
$\ell = \mathsf{K}, B\ell = N, \mathsf{c} = \mathsf{K}/2, K = \sqrt{\mathsf{N}}$	quad	$\mathsf{N} \in \Theta(\eta^2)$	quad	$\mathsf{N} \in \Theta(\eta^2)$	strong	$p > \frac{2\eta}{\mathsf{K}}$	strong	$p > \frac{2\eta}{\mathsf{K}}$
$\ell = \eta, B\ell = N, \mathsf{c} = \mathsf{K}/2, \mathsf{K} \in O(1)$	none		none		strong	$p > \frac{\eta}{\ell - \mathsf{c}}$	strong	$p > \frac{\eta}{\ell - \mathsf{c}}$
$\ell = \mathsf{K}, B\ell = N, \mathsf{c} = \mathsf{K} - 1$	quad	$\mathsf{K} \geq \eta^2$	strong		strong	$p \geq \frac{\eta}{\mathsf{K}-1}$	strong	$p \geq \frac{\eta}{\mathsf{K}-1}$
$\ell = \mathsf{K}, B\ell = N, \mathsf{c} = \mathsf{K} - 1$	quad	$\mathsf{K} \geq \eta^2$	strong		quad	$p < \frac{1}{\mathsf{K}-1}$	quad	$p < \frac{1}{\mathsf{K}-1}$
$\ell = \mathsf{K} - 1, B\ell = N, \mathsf{c} = \mathsf{K} - 1$	quad	$\mathsf{K} \geq \eta^2$	quad		quad	$p < \frac{1}{\mathsf{K}-1}$	quad	$p < \frac{1}{\mathsf{K}-1}$
$\ell = \mathsf{K} - 1, B\ell = N, \mathsf{c} = \mathsf{K} - 1$	quad	$\mathsf{K} \geq \eta^2$	quad		strong	$p \geq \frac{\eta}{\mathsf{K}-1}$	strong	$p \geq \frac{\eta}{\mathsf{K}-1}$
$\ell = \mathsf{K} - 1, B\ell = 2N, \mathsf{c} = \mathsf{K} - 1$	quad	$\mathsf{K} \geq \eta^2$	strong		strong	$p \geq \frac{\eta}{\mathsf{K}-1}$	strong	
$B = \sqrt{\mathsf{N}}, \ell = \sqrt[4]{\mathsf{N}}, \mathsf{c} = \sqrt{\mathsf{K}}$	none		none		strong	$p \geq \frac{\eta}{\sqrt{\mathsf{K}}}$	strong	$p \geq \frac{\eta}{\sqrt{\mathsf{K}}}$
$B = \sqrt{\mathsf{N}}, \ell = \sqrt{\mathsf{N}}, \mathsf{c} = \sqrt{\mathsf{K}}$	quad	$N, K \in \Omega(\eta^2)$	quad	$N, K \in \Omega(\eta^2)$	strong	$p \geq \frac{\eta}{\sqrt{\mathsf{K}}}$	strong	$p \geq \frac{\eta}{\sqrt{\mathsf{K}}}$
$B = \sqrt{\mathsf{N}}, \ell = \sqrt{\mathsf{N}}, \mathsf{c} = log(\mathsf{K})$	none		none		none		none	
$B = \frac{\mathsf{N}}{\sqrt{\eta}}, \ell = \sqrt{\eta}, \mathsf{c} = 2\eta, \mathsf{K} = 4\eta$	quad		quad		strong	$p > 1/2$	strong	$p > 1/2$
$\frac{\mathsf{K}}{\mathsf{c}} = 2, B = 0, \ell = log(\eta)$	none		none		none		none	
$\frac{\mathsf{K}}{\mathsf{c}} = 2, B < \frac{\mathsf{N}}{2log(\eta)}, \ell = log(\eta)$	none		none		none		none	
$\frac{\mathsf{K}}{\mathsf{c}} = 2, B \geq \frac{2\mathsf{N}}{log(\eta)}, \ell \geq log(\eta)$	none		quad		none		strong	
$\frac{\mathsf{K}}{\mathsf{c}} = 4, B > \frac{\mathsf{N}}{log(\eta)}, \ell \geq log(\eta)$	quad		strong		quad		strong	

even for the recipients.

XI. Conclusion and Future Work

In this work we show that protocols with secret-sharing have better hopes for anonymity. We motivate the protocol designers to build new protocols in that direction. Even then the anonymity of the protocols will be bounded by the impossibility conditions presented in this paper, unless there exist a protocol that can efficiently break our assumptions on secret sharing. In case, a protocol finds an efficient way to achieve mixing in a dishonest node, still the protocol will be restricted by the condition $\ell p > 1$ for strong anonymity. That leaves us with the other assumption on secret sharing, and the following question:

> Is it possible to build an AC protocols that uses a secret sharing scheme that generates only $w < k \times n$ shares for n messages where at least k shares are necessary to reconstruct all the m messages correctly, without using any trusted third party, with a communication of $O(n)$ and constant latency overhead?

If such a protocol can exist, that protocol will escape the impossibility conditions provided in this paper. By proving impossibility in all other direction, we show the plausible path to the research community.

References

[1] 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.

[2] M. Ando, A. Lysyanskaya, and E. Upfal, "On the complexity of anonymous communication through public networks," *CoRR*, vol. abs/1902.06306, 2019. [Online]. Available: http://arxiv.org/abs/1902.06306

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

[4] 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.

[5] D. Lazar and N. Zeldovich, "Alpenhorn: Bootstrapping secure communication without leaking metadata," 10 2016.

[6] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, vol. 4, no. 2, pp. 84–88, 1981.

[7] D. I. Wolinsky, H. Corrigan-Gibbs, B. Ford, and A. Johnson, "Dissent in Numbers: Making Strong Anonymity Scale," in *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, 2012, pp. 179–182.

[8] 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.

[9] H. Corrigan-Gibbs and B. Ford, "Dissent: Accountable Anonymous Group Messaging," in *Proc. 17th ACM Conference on Computer and Communication Security (CCS)*, 2010, pp. 340–350.

[10] 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.

[11] T. K. Srikanth and S. Toueg, "Simulating authenticated broadcasts to derive simple fault-tolerant algorithms," *Distributed Computing*, vol. 2, no. 2, pp. 80–94, 1987.

[12] R. Gennaro, M. O. Rabin, and T. Rabin, "Simplified VSS and fact-track multiparty computations with applications to threshold cryptography," in *Proc. ACM PODC*, 1998, pp. 101–111.

[13] D. Das, S. Meiser, E. Mohammadi, and A. Kate, "Anonymity trilemma: Strong anonymity, low bandwidth, low latency—choose two," Cryptology ePrint Archive, Report 2017/954, 2017, https://eprint.iacr.org/2017/954.

[14] 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.

[15] 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.

[16] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi, "AnoA: A Framework For Analyzing Anonymous Communication Protocols," in *Proc. 26th IEEE Computer Security Foundations Symposium (CSF 2013)*, 2013, pp. 163–178.

[17] 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)*, vol. 7(2), no. 5, 2016.

[18] 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.

[19] 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.

[20] ——, "On the Complexity of Anonymous Communication Through Public Networks," *CoRR arXiv*, vol. abs/1902.06306, 2019.

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

[22] 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.

[23] D. Chaum, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *Journal of Cryptology*, vol. 1, no. 1, pp. 65–75, 1988.

[24] S. Goel, M. Robson, M. Polte, and E. Sirer, "Herbivore: A scalable and efficient protocol for anonymous communication," 2003.

[25] A. Kwon, D. Lazar, S. Devadas, and B. Ford, "Riffle: An efficient communication system with strong anonymity," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 2, pp. 115–134, 2016.

Appendix A
Protocol Model Revisited.

A. Construction of a Concrete Adversary

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

$$
\begin{aligned}
S_j = \{p = (&t_1.\mathsf{prev}, \ldots, t_k.\mathsf{prev}, t_k.\mathsf{next}) : \\
&((t_1, r_1), \ldots, (t_k, r_k)) \in \mathsf{Tokens\ s.t.}\ k \leq \ell \\
&\wedge t_1.\mathsf{prev} = u_j \wedge t_k.\mathsf{next} = R \wedge t_k.\mathsf{msg} = \texttt{Chall} \\
&\wedge \forall_{i \in \{1,\ldots,k-1\}} (t_i.\mathsf{next} = t_{i+1}.\mathsf{prev} \wedge r_{i+1} = r_i + 1 \\
&\wedge (\ \exists t'_{i+1} : (t'_{i+1}, r_{i+1}) \in \mathsf{Tokens}\ \ \wedge t'_{i+1}.\mathsf{prev} = t_i.\mathsf{next} \\
&\qquad \wedge t'_{i+1}.\mathsf{ID_t} = t_i.\mathsf{ID_t}) \Rightarrow t'_{i+1} = t_{i+1})\}
\end{aligned}
$$

Definition 2 (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.*

$\mathcal{A}_{paths}$ thus checks whether both challenge users *could have* sent the challenge message, and explicitly ignore differences in probabilities of the challenge users having sent the challenge message, as those probabilities can be protocol specific. Naturally, when $\mathsf{c} = 0$, $\mathcal{A}_{paths}$ represents a *non-compromising* (yet global network-level) adversary, i.e., an adversary that compromises no protocol nodes but eavesdrops all links between nodes; but when $\mathsf{c} \neq 0$, $\mathcal{A}_{paths}$ is *partially compromising*.