Mixnets on a tightrope: Quantifying the leakage of mix networks using a provably optimal heuristic adversary

Sebastian Meiser *University of Luebeck*
Debajyoti Das *KU Leuven*
Moritz Kirschte *University of Luebeck*
Esfandiar Mohammadi *University of Luebeck*
Aniket Kate *Purdue University / Supra Research*

***Abstract*—Mixnets are widely believed to hide communication metadata of individuals. We show that there are various pitfalls when designing mixnet topologies and routing strategies, in particular when choosing mixnets with low delays. We introduce a tool that empirically evaluates such a leakage in mixnets and show that this tool precisely estimates this leakage for recipient anonymity, up to an error introduced by sampling. First, we introduce a novel generic attack strategy that we even prove to be optimal for breaking recipient anonymity. In contrast to prior work, our attack strategy incorporates the severity of each observation's leakage, via its so-called privacy loss. Second, our tool provides a lower bound on an attacker's advantage against recipient anonymity by sampling a large set of observations; if a significant number of observations with high privacy loss is observed, the tool outputs a lower bound on the leakage by providing a lower bound on the mass of the tail of the distribution of privacy losses. From the literature, we study the topology and routing strategies of the Karaoke and Atom protocols, provide bounds on their leakage and recommend design choices based on the analysis.**

1. Introduction

Mixing networks (or mixnets) [8] aim to hide the communication metadata of its users by letting all messages follow a cascade of mixing nodes (or mixnodes), each of which shuffles those messages. In practice, as the number of messages grows, relying on the cascade approach does not scale in terms of (cryptographic) computation and communication costs; thus, most mixnet designs distribute the traffic load horizontally over several nodes to scale for a large number of messages. Particularly, mixnets follow different routing strategies such as square-lattice shuffling [20], butterfly network [1], or stratified network [21], [22], [27]. While such a communication strategy seems to hide communication metadata, it is not clear which network topologies and routing strategies are necessary to achieve strong anonymity properties. In particular, prior analyses of mixnets were highly untight and sometimes even wrong [13].

Prior analyses of mixnets did not consider that the probability for a given adversarial metadata observation to occur is higher for some user actions (say, "Alice talks to Bob") than for other user actions (say, "Alice talks to Charlie"). Consider recipient anonymity: the attacker spies on communication metadata and wants to find out whether Alice sent a message to Bob or Charlie. For a given metadata observation, the probability might be non-zero that Alice sent a message to Bob and non-zero that Alice sent a message to Charlie; yet, the probabilities of these two events might be vastly different (see Figure 1). We call the ratio of these probabilities for a given observation the *a posterori biases* (or its logarithm the privacy loss). This raises the question: "Can such a posteriori biases be successfully utilized by attackers against anonymity properties of mixnets?"

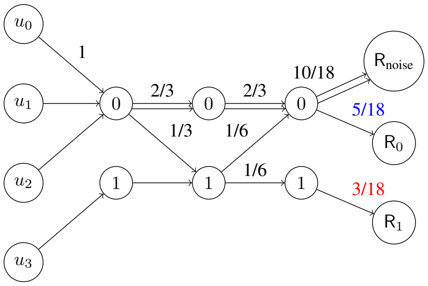


Figure 1: Example observation annotated with the probabilities as propagated by the adversary. There are more randomness tapes that lead to this observation if u_0 sends a message to R_0 than if u_0 sends a message to R_1. For simplicity of understanding, none of the nodes are compromised. For this simplified example, there are exactly 8 ways to get this observation; 5 if u_0 sends the message to R_0 and 3 if u_0 sends the message to R_1. We list those in Appendix 2.

Prior work takes initial steps for empirically estimating leakage via a posteriori biases [2], [6], [7]. Yet, this prior work solely quantified a lower bound on the a posteriori bias. This prior work has not estimated a lower bound on the adversarial advantage, which is typically estimated in anonymity analyses.

Our contribution. We positively answer this question for recipient anonymity and show that mixnets with low latency have significant leakage. Our contribution is threefold.

1) We introduce an optimal attacker against recipient anonymity that precisely computes the probability that, for a given observation, Alice sent a message to Bob and the probability that Alice sent a message to Charlie. In particular, the optimal attack is able to take a posteriori

biases into account by computing the so-called privacy loss, i.e., the logarithm of the ratio of probabilities.

2) We utilize this optimal attacker to construct a tool to analyze these a posteriori biases. Our results show that taking these a posteriori biases can lead to an order of magnitude increased adversarial advantage compared to restricting the analysis to observations where the probability of sending a message to Bob or Charlie is 0. This tool samples mixnet meta-data observations and uses them to estimate an upper bound and lower bound on the attacker's advantage δ, with high confidence. In contrast to entropy-based empirical evaluation [14], our tool provides provable and quantifiable confidence intervals on δ. We detail their limitations and compare them with us in Section 2.3.

It might be of independent interest that we derive a novel and robust empirical measure of leakage that is tighter than prior empirical estimates and not only applicable to the analysis of anonymity properties but also to security properties and differential privacy.

3) We study two major protocols, Karaoke [22] and Atom [20]. Prior proofs attempt to identify sufficient conditions, which we call 'gadgets', that have turned out to be insufficient. We introduce a GadgetTester that samples observations and reports the maximally observed a posteriori bias that satisfies the gadget. The GadgetTester can thus quickly generate good counter-examples against a gadget-based proof. Furthermore, we report insights towards improving their privacy and scalability and generalize them for Karaoke's successors Yodel [23], and Groove [5].

Key insights. Our tool uses our optimal attack strategy (Section 3.2) and evaluates the anonymity guarantees of several existing mixnets, as well as providing general insights. We quantitatively confirm some insights provided by the existing anonymity trilemma results [10], [12]. Additionally, taking a posteriori biases into account leads to far tighter lower bounds (often by about one order of magnitude) than just quantifying the probability of a total breakdown of anonymity. In particular, a stratified mixnet with three hops has significant leakage, and the a posteriori bias is so high that it does not even provide meaningful deniability in the sense of differential privacy.

We show that the proof technique used by Karaoke [22] is inaccurate; and therefore, the anonymity guarantees of Karaoke and related designs [5], [23], [29] might be flawed. More specifically, their proof technique attempts to identify a *sufficient conditions* for mixing (cf. Section 2.1). Yet, their proof does not take a posteriori biases into account and leaks information to the adversary based on the number of messages transmitted through the network links. Our evaluation of their end-to-end protocol provides evidence for differential privacy-like guarantees. Additionally, we show directions towards making Atom-like designs more practical by relaxing its communication model and network topology. We provide detailed insights about them in Section 7.

Symbol	Description
N	number of users
R	number of rounds
k	width of the network
y	privacy loss
∇	instance of randomness tapes
$\mathcal{N}$	set of nodes (per layer)
C	set of compromised nodes (per layer)
$t_{i,k}$	encrypted packet on the k-th hop from sender u_i
$t_{i,k}.prev$, $t_{i,k}.next$	two nodes of the k-th hop of $t_{i,k}$
u_0	sender Alice
R_0, R_1	the two potential recipients
scenario A	the case where R_0 receives the message
scenario B	the case where R_1 receives the message

TABLE 1: Notation Table

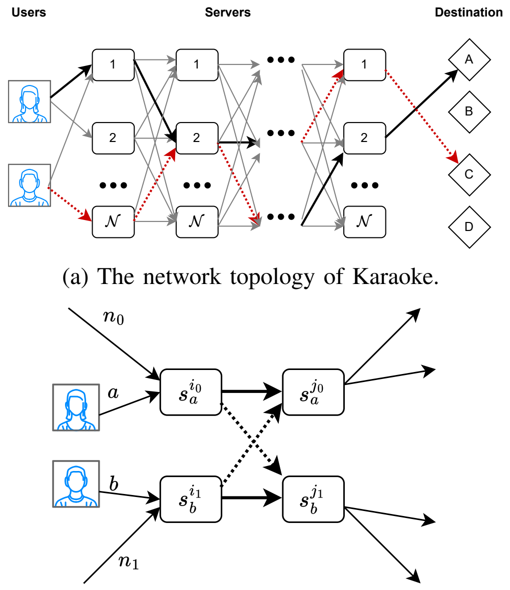
2. Background and related work

2.1. Relevant mixnet designs and motivation

Several approaches exist in the literature to make mixnets scalable based on different network topologies and different routing strategies. The protocols that attempt to achieve provable guarantees under indistinguishability-based notions (how easily the adversary can distinguish who among Alice and Bob could have sent a certain message) typically follow a proof strategy similar to the following: (i) identify a set of conditions or *'gadget'* that are sufficient to ensure the shuffle of the two concerned messages (e.g., the messages from Alice and Bob), (ii) compute the probability of that *gadget* appearing on the paths of the messages. Depending on the routing strategy, topology, etc., it is not always straightforward to construct such gadgets, and many times badly constructed gadgets could lead to incorrect security proofs. We summarize below some of the relevant protocols, highlighting the gadgets used in their proof technique and why those gadgets are incomplete.

2.1.1. Karaoke and Stadium. Karaoke [22] and Stadium [29] (and their successors Yodel [23] and Groove [5]) can scale to millions of users by utilizing multiple paths in a layered topology (c.f. Fig. 2: for each hop/layer of a message, the sender chooses a mixnode from the set of all mixnodes in the system. To achieve link saturation, they leverage numerous noise messages in $\Theta(|\text{servers}|^2)$, which yields a property similar to relationship anonymity. Additionally, they require the clients to send messages in batches; each client sends exactly one message in each batch, and whenever they don't have a real message to send, they send a dummy message to complete a batch.

For the security argument, Karaoke uses the following gadget: let a and b are two honest messages, and n_0 and n_1 are two noise messages generated by honest parties (users or servers). The messages a and n_0 both go through the same honest server/mixnode $s_a^{i_0}$ at layer i_0, and the messages b and n_1 meets at an honest mixnode $s_b^{i_1}$ at layer i_1. Then, b meets n_0 at layer j_1 at an honest mixnode $s_b^{j_1}$, and a



(a) The network topology of Karaoke.

(b) The *gadget* used in their anonymity proof.

Figure 2: The routing strategy and *gadget* of Karaoke.

meets n_1 at layer j_0 at an honest mixnode $s_a^{j_0}$, such that $\max\{i_0, i_1\} < \min\{j_0, j_1\}$. They claim that the above event is indistinguishable from the event where a gets swapped with n_0 at layer i_0 and b gets swapped with n_1 at layer i_i, then a and b meet n_1 and n_0 at layers j_1 and j_0, respectively.

However, this proof does not consider that different links can have a different number of messages, and that could make some observations more likely compared to others. Das et al. [13, Appendix A.1] have shown the existence of a counter-example where the above gadget is satisfied even though the protocol has significant leakage in those configurations. That demonstrates the necessity of a thorough verification of the anonymity guarantees (or the gadgets which the proof is relying on) for a concerned protocol. We have presented a similar case in Figure 1, where a bias exists that can be exploited. To analyze such gadgets and find counter-examples we have implemented a simple gadget tester (c.f. Section 6.2) for the gadget shown in Figure 2b.

The main difference between Stadium [29] and Karaoke is in how they defend against active attacks: Stadium uses verifiable shuffle to detect and defend against packet drops by the mixnodes; on the other hand, Karaoke uses Bloom filters to detect such packet drops.

The protocols Yodel [23] and Groove [5] follow a similar mixing strategy, except that they maintain persistent connections or *circuits*. Users maintain their circuits for the same amount of time (similar to *epochs*), using them for applications like voice calls. A global adversary can anyway link together all the packets related to the same circuit (since they are part of the same voice call session). However, their goal is to obfuscate the two ends of a specific circuit.

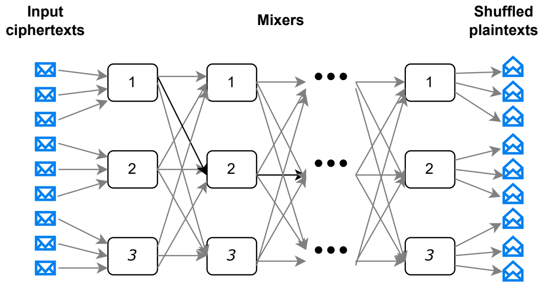


Figure 3: The square network topology of Atom deployed with $\mathcal{N} = 3$ servers and $N = 9$ messages in a batch.

2.1.2. Atom. Atom [20] can provide strong anonymity guarantees by deploying a square network topology (as shown in Fig. 3: for a batch of N messages, the system requires are $\mathcal{N} = \sqrt{N}$ servers. In a given round, each server receives a subset of exactly $\sqrt{N}$ messages, shuffles that subset, and equally distributes the messages among all the servers for the next round. If all the servers are honest, based on Hastad's square shuffle analysis [19], after $T > 10$ rounds of such iterations the output would be indistinguishable from a uniform permutation.

However, not all servers could be honest in practice: they suggest using a cluster of 32 servers that collectively shuffle a subset for a given iteration: as long as any one of those 32 servers is honest, the cluster would emulate an honest mixer with all but a negligible probability. This puts a significant deployment overhead for such designs to be used effectively. It is natural to ask: what would the anonymity guarantee when the clusters are significantly smaller in size so that there is a small but non-negligible probability that a cluster could be completely compromised? The existing analysis (in [19]) is not applicable anymore.

In practice, enforcing the exact same number of messages received by each mixing cluster is not easy. Especially if the design lets users choose the path of their messages, such assumptions will not hold anymore. It is not yet known how to prove the anonymity guarantees formally for Atom-like designs with such a communication model. Our tool allows a thorough analysis of the guarantees of such systems, with a tolerable error margin.

2.2. Analysis tools for a posteriori biases

The literature contains some results for empirically estimating a lower bound on leakage that takes a posteriori biases into account [2], [6], [7]. This work concentrated on differentially private mechanisms and solely quantified a lower bound on the a posteriori bias as a multiplicative error of the attacker, called a bound on the so-called privacy loss or ε. In contrast to our work, this prior work does not take into account how often and which kind of bias occurs and has thus not estimated a lower bound on the adversarial advantage, which is typically estimated in anonymity analyses.

2.3. Experimental evaluation based on entropy

Evaluation of anonymity based on entropy [14] also requires estimating the probabilities correlating the input and output messages of the mixnet, for a chosen instance of that mixnet. That is similar to executing our adversary for a single observation of the mixnet protocol. Even our adversary (defined in Section 3.1) has strong similarities with the adversary considered in [14] (and its implementation Mixim [18]).[1] However, there are some subtle differences. Our tool/adversary first samples an observation, and then computes the chance(s) that the observation is produced by either the message being linked to Alice or to Bob, based on all possible random choices in the protocol (choice of the users, mixnodes, etc). On the contrary, for experimental evaluation of entropy [18], the random choices (of the users and other parties in the protocol) are controlled in the experiment, and then the observation is generated and the probability that the adversary can track a specific target message is computed.

Our tool can accurately verify when a *gadget* is badly designed (cf. Section 6.2). Evaluating such gadgets is not possible with Mixim. In the absence of such a gadget, when we need to evaluate the end-to-end protocol, both methods produce some sampling error. However, because of our formal approach to defining the adversary, we can quantify and bound the sampling error. It is not straightforward to quantify the overall error for entropy-based evaluations.

Our proof of optimality for the adversary provides strong evidence for the soundness of the adversary used by entropy-based techniques. However, our proof shows the optimality only for recipient anonymity, which is an indistinguishability-based notion.

```
Adaptive AnoA Challenger $\textsc{Ch}(\Pi, \alpha, \gamma, b)$

Initialize the Game
  Set $\textsc{Sessions} := \emptyset$, $\mathsf{States} := [\texttt{FRESH}, \texttt{FRESH}, \ldots, \texttt{FRESH}]$ ($\gamma$ entries, one per challenge).

Upon message $\texttt{Input} = (\mathsf{S}, \mathsf{R}, m, \mathsf{z})$
  $\textsc{SimulateProtocol}(\Pi, \mathsf{S}, \mathsf{R}, m, (\mathsf{A}, \mathsf{z}))$

Upon message $\texttt{Challenge} = (\mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, m, \Psi)$
  if $\Psi \notin \{1, \ldots, \gamma\} \vee \mathsf{States}[\Psi] = \texttt{OVER}$ then
    Output $\bot$.
  else
    $(\mathsf{state}', \mathsf{S}^*, \mathsf{R}^*) \leftarrow \alpha(\mathsf{States}[\Psi], \mathsf{S}_0, \mathsf{S}_1, \mathsf{R}_0, \mathsf{R}_1, b)$
    Set $\mathsf{States}[\Psi] := \mathsf{state}'$.
    $\textsc{SimulateProtocol}(\Pi, \mathsf{S}^*, \mathsf{R}^*, m, (\textsc{Ch}, \Psi))$
  end if

Upon message $(m, \textsc{sid})$ from $\Pi$
  if $(\mathsf{ID}, \textsc{sid}) \in \textsc{Sessions}$ for some value $\mathsf{ID}$ then
    Send $\texttt{Observations} = (m, \mathsf{ID})$ to $\mathsf{A}$.
  end if

Subroutine SimulateProtocol$(\Pi, \mathsf{S}, \mathsf{R}, m, \mathsf{ID})$, where $\mathsf{ID} = (\textsc{Ch}, \Psi)$ or $\mathsf{ID} = (\mathsf{A}, \mathsf{z})$
  if $(\mathsf{ID}, \textsc{sid}) \in \textsc{Sessions}$ for some value $\textsc{sid}$ then
    Retrieve this $\textsc{sid}$
  else
    Set $\textsc{sid} \leftarrow \mathcal{N}$
    Store $(\mathsf{ID}, \textsc{sid})$ in $\textsc{Sessions}$
  end if
  Run $\Pi$ on $(\mathsf{S}, \mathsf{R}, m, \textsc{sid})$.
```

Figure 4: The full AnoA challenger from Backes et al. [3], included for completeness.

3. Definitions: Anonymity, Adversary

3.1. Game definition

In this paper, we leverage the AnoA framework for anonymity [3]. Here, anonymity is defined in terms of an indistinguishability game between a challenger that runs the protocol and an adversary that chooses the user behavior. The adversary eventually sends a challenge message that has a user send a message to one of two potential recipients. The Challenger (cf. Figure 4) selects one of those based on its secret challenge bit and continues to run the protocol. The goal of the adversary is to guess the secret challenge bit.

In this work we focus on recipient anonymity as represented in AnoA and formally define an adversary class, to ensure that the adversary fits to our scenario.

Our protocol's wrapper and adversary class. Our combination of protocol wrapper adversary class ensures that every sender sends a message and every recipient receives a message. All those messages from senders are sent to an unimportant recipient $\mathsf{R}_{\text{Noise}}$ where they can be discarded and the messages for our two recipients of interest originate in unimportant senders. Formally, we require that $\mathsf{R}_{\text{Noise}}$ receives a message from every sender. The actual challenge message then replaces one of those messages: Alice's message is sent to either Charlie or Dave, whereas a random other sender is selected to send a message to the one Alice doesn't send a message to. This is to ensure that the adversary cannot from the mere fact that Dave receives a message infer that he was the challenge recipient.

In terms of AnoA's sessions, we treat every session as an individual run of the protocol. Our results thus compose sequentially as by AnoA's single-challenge-reducibility. However, they do not compose in parallel: when two challenge messages might interfere, our adversary is not necessarily optimal anymore. We refer to Section 7.4 for a discussion.

3.2. Definition of the adversary

Below we present an efficient way to compute and capture the observations that a global passive adversary can make when looking at interactions within a stratified mix network — we call this adversary heuristic adversary.

[1] They do not have an explicit definition of an adversary; however, it is implicit in their model.

These observations include packets from users to nodes as well as packets between nodes. They do not include any cryptographic aspects of the protocols in question. We first present a formal definition of an optimal adversary that takes into account all possible cases that could lead to a certain observation. We then show that our heuristic adversary is equally strong.

Network topology. For simplicity of explanation, we consider a stratified network topology where mixnodes are arranged in ℓ ordered layers. The layers are interconnected such that each mixnode in layer i receives messages from mixnodes in layer $i-1$ and sends messages to mixnodes in layer $i+1$; while the first layer receives messages from senders and the last layer forwards messages to their final recipients. The path length of message routes is determined by the number of layers ℓ. To select a message route, each user chooses the nodes for each message uniformly at random from each layer. Although we consider a stratified topology our analysis is also valid for a free routing topology where the users can choose each hop of the path from all available mixnodes in the whole mixnet.

Client/Game Setting. We aim to analyze the 'mixing' quality of a mixnet against our heuristic adversary. We don't want to measure or model scenarios where the adversary could win even if the mixnet was a trusted third party (e.g., if only one of the recipients ever received a message). To that end all of our N senders send exactly one message each and both of our recipients receive one message each.

3.3. Privacy loss

A notion closely related to our analysis is the so-called *privacy loss* used in the field of privacy-preserving mechanisms and personal data protection. Closely related to *differential privacy*, the privacy loss of an event is the (logarithm of the) ratio between probabilities to observe the event in one case, divided by the probability to observe the same event in the other case. Applied to our notion of recipient anonymity, the privacy loss is the logarithm between the probability of observing if the target recipient is R_0 divided by the probability of making the same observation if the target recipient is R_1 (in differential privacy called x_0, x_1). In cases where the numerator is zero, the privacy loss is zero (and the logarithm $-\infty$); in cases where the denominator is zero, the privacy loss (and its logarithm) is ∞.

Definition 1 (Privacy loss)**.** *Let $M\colon \mathcal{X} \mapsto \mathcal{U}$ denote a probabilistic measure, $x_0, x_1 \in \mathcal{X}$ some inputs, some $a, b \in \{0,1\}$, and $o \in \mathcal{U}$ an observation from universe $\mathcal{U}$.*

Then $\mathcal{L}_{x_{a:b}}\colon \mathcal{U} \mapsto \mathcal{Y}$ denotes the privacy loss with support $\mathcal{Y} = \bigcup_{o\in\mathcal{U}} \{\mathcal{L}_{x_{a:b}}(o)\} \subset \mathbb{R} \cup \{-\infty, \infty\}$ s.t. $\mathcal{L}_{x_{a:b}}(o) =$

$$\begin{cases} \ln \frac{\Pr[M(x_a)=o]}{\Pr[M(x_b)=o]} & \textit{if} \forall_{\eta\in\{0,1\}}\colon \Pr[M(x_\eta)=o] \neq 0 \\ \infty & \textit{else if } \Pr[M(x_b)=o]=0 \\ -\infty & \textit{else.} \end{cases}$$

The PDF of the privacy loss distribution (PLD) for each atomic event $y \in \mathcal{Y}$ is given by

$$\omega_{M,x_{a:b}}(y) = \sum_{\left\{o \mid \mathcal{L}_{x_{a:b}}(o)=y, o\in\mathcal{U}\right\}} \Pr[M(x_a)=o].$$

For completeness and to discuss the meaning of the respective terms, we introduce the definition of (approximate) differential privacy and discuss it

Definition 2 (Differential Privacy)**.** *Let $M\colon \mathcal{X} \mapsto \mathcal{U}$ denote a probabilistic measure. M is (ε,δ)-differentially private if for all* neighboring *inputs $x_0, x_1 \in \mathcal{X}$ and for all sets $S \subseteq \mathcal{U}$ ob observations from universe $\mathcal{U}$ we have:*

$$\Pr[M(x_0) \in S] \leq e^{\varepsilon} \Pr[M(x_1) \in S] + \delta$$

For an $\varepsilon \geq 0$, we write $\delta(\varepsilon)$ to denote the smallest value for δ such that M is $(\varepsilon, \delta(\varepsilon))$-differentially private.[2]

If M is $(\varepsilon, 0)$-differentially private, we call it ε-differentially private. In this case, the above definition of approximate differential privacy reduces to what is sometimes called pure differential privacy.

In either case, ε measures the degree of bias that we allow in observations, with a higher value of ε corresponding to a higher allowed bias. For our purpose, we cannot use pure differential privacy, as for any mix network with a width ≤ 2 there will always be an observation where the challenge message never mixes with any other message. This probability tends to be exceedingly unlikely, but this unlikelihood needs to be captured within δ.

Generally, δ captures two things: The probability of distinguishing events in which an attacker can trivially win the game, and a probability mass that exceeds the bias described by ε. If we allow for a bias of e^{ε} and an unlikely event occurs in $M(x_0)$ with a probability of p_0 and a bias of e^y with $y > \varepsilon$, then a fraction of the probability, namely $(1-e^{\varepsilon-y}) \cdot p_0$ contributes to δ. This is exactly the probability mass that is outside of the bound set by ε.

Formally, we define the ε-attack advantage $\delta(\varepsilon)$ based on the privacy loss (cf. Definition 1) as follows where for $\varepsilon = 0$ we obtain the unbiased attack advantage. Following [24], the tightest δ for a given ε can be computed as follows.

Definition 3 (ε-Attack Advantage, cf. [28], Definition 6)**.** *Let $\omega_{M,x_{a:b}}$ denote a PLD as in Definition 1 with support $\mathcal{Y} = (y_i)_{i=1}^{m}$ s.t. $y_i \leq y_{i+1}$ (ascending-sorted). For $\varepsilon \geq 0$, we define the ε-attack advantage δ:*

$$\begin{aligned} \delta(\varepsilon, \omega_{M,x_{a:b}}) = & \sum_{y_i \in \mathcal{Y}\setminus\{\infty\}} \max(0, (1-e^{\varepsilon-y_i})\, \omega_{M,x_{a:b}}(y_i)) \\ & + \omega_{M,x_{a:b}}(\infty) \\ \delta'(\varepsilon, \omega_{M,x_{0:1}}, \omega_{M,x_{1:0}}) = & \max_{(a,b)\in\{(0,1);(1,0)\}} \delta(\varepsilon, \omega_{M,x_{a:b}}) \end{aligned}$$

[2] Note that for any M and for any $\varepsilon \geq 0$ we can always find a value for δ such that M is (ε,δ)-differentially private. Even for completely broken M we can fulfill the inequality with $\delta = 1$; for reasonable M we will have $\delta \ll 1$.

4. Optimality of the heuristic adversary

When computing what occurs in the network, we can fully capture every aspect of metadata available to the adversary by considering the random path choices of the parties involved. We can model those by considering each sender to have an independent randomness tape (unavailable to the adversary) that dictates all these choices. All instances of those tapes are equally likely and the sum total of them covers everything that could possibly happen in our protocol run (since we exclude network issues or computer failures).

The game looks as follows: given an observation, we want to find out what the probability is that the message of Alice ended up in a specific node. A successful adversary tries to trace Alice's message optimally to compute for each node the precise probability that Alice's message ended up in that node. For recipient anonymity, the adversary can then directly guess which recipient received her message.

The *optimal adversary* acts as follows: given an observation, the adversary iterates over all randomness tapes. For each instance of randomness tapes, the adversary runs the idealized version of the protocol $Prot_{\text{ideal}}$ on those random coins to compute an observation and a state of the messages after the protocol run. It then checks whether the observation fits the target observation and whether Alice's message ends up in the chosen node. If so, the adversary increases the counter for that node by one (all counters initialized at zero).

At the end of this, the adversary has a counter c_n for every node n and then outputs a list of probabilities as $[\frac{c_n}{\sum_{j \in \mathcal{N}} c_j}$ for $n \in \mathcal{N}]$. Note that given a scenario (who sends messages to whom) and a network architecture, $\sum_{j \in \mathcal{N}} c_j$ specifies the number of instances that lead to the given observation. With $\frac{\sum_{j \in \mathcal{N}} c_j}{\#\text{allobservations}}$ we have the probability to yield this specific observation.

Definition 4 (Optimal recipient anonymity). *We call an adversary* A *optimal for the recipient anonymity game if its outputs match the optimal adversary.*

4.1. (Partial) Instances

A key insight is that we can relate (partial information about) the randomness tapes directly to relevant aspects of the observation.

Definition 5 (partial instance). *A partial instance is a set of instances with common elements and parts where they differ, denoted by $*$. For example, a partial instance $\nabla = [*, 1, *, *, \dots, *]$ contains all instances that have a "1" at their second position.*

Given partial information about one or more nodes, i.e, which messages are in the nodes and what the edge weights are from those nodes, we say that ∇ is a partial instance consistent with the partial information, if all instances I in ∇ are consistent with the partial information and if there are no instances $I \notin \nabla$ that are also consistent with the partial information. We call ∇ a partial instance about a node i in round r if it contains values different from $*$ for rounds r and $r+1$ of messages that in round r are in node i, and contains $*$ everywhere else.

By their nature, partial instances group together all instances consistent with them.

We characterize the information known to the adversary from observing the network as an *observation*. For now we define observations solely by their meta-data from observing connections between network participants.

Definition 6 (Observation). *An observation contains for every round r the number of messages sent by each sender or node to each other node or recipient. Formally, we consider an observation as a graph, where each graph node is a network node in a given round and each edge denotes messages being sent from this node to other nodes. If the network topology is not a cascade, then we use several graph nodes (one for each round) to depict the same network node. As each graph node depicts a network node in a specific round, we denote it with the number of messages inside that node in the given round. Each edge has a weight equal to the number of messages being sent. The sum of edge weights leaving each node naturally corresponds to the number of messages in a node in a given round. If a network node holds on to k messages from round r to round $r+1$, then the corresponding graph node for round r has an edge with weight k to the graph node corresponding to the same network node, but for round $r+1$.*

Lemma 1 (number of partial instances in relation to edge weights). *For any node i that in a round r has n messages inside it, with outgoing edges with weights $w_1, \dots, w_k$, the number of partial instances of node i in round r consistent with those n messages in the node and those edges is given by $\frac{n!}{\prod_j w_j!}$.*

We refer to Appendix 1.1 for the proof. Lemma 1 boils down to saying: There are $\frac{n!}{\prod_j w_j!}$ ways to distribute the n messages in a node i over the k edges with weights $w_1, \dots, w_k$ and this directly corresponds to partial instances for node i in that round.

Very important to the overall proof approach is that the probabilities derived from partial information on disjoint nodes are independent and the numbers of partial instances consistent with them can be multiplied with each other: Given two pieces of partial information, say, information about the edge weights for two different nodes i and j in the same round. If there are p_i partial instances consistent with the information about i and p_j many partial instances consistent with the information about j, then there will be $p_i \cdot p_j$ many partial instances consistent with the combined information about i and j. In other words, information about the flow of messages from node i is not dependent on the flow of messages from node j. This property and its inherent link to our adversary's performance is why we focus on single-message anonymity notions.

Lemma 2 (Node edge weights indicate probability factors). *We assume that the choice of each node in each round is performed via uniformly distributed choice over all possi-*

ble nodes for that round and is independent of any prior choices for the same path and of any choices made for other messages. For any node i with n messages inside, if there is an edge with weight w_x to node x then out of all partial instances consistent with the edge weights from node i that have the message m in node i, a fraction $\frac{w_x}{n}$ will subsequently have m in node x.

We refer to Appendix 1.2 for the proof.

4.2. The heuristic adversary

The heuristic adversary $\mathrm{A}_{\mathrm{Heu}}$ tracks probabilities for messages in nodes and via edges between them. We define the heuristic adversary as follows.

Definition 7 (Heuristic adversary)**.** *Given an observation as in Definition 6, the one-message heuristic forward adversary without output $\mathrm{A}_{\mathrm{forward}}$ keeps a table of probabilities p_i^r per round r and per node i.*

Initialization: If the challenge sender u_0 sends their challenge message to node i in round $r = 0$, then $\mathrm{A}_{\mathrm{forward}}$ assigns the probability $p_i^0 = 1.0$ and assigns for all nodes $j \neq i$ the probability $p_j^0 = 0.0$.

Subsequent rounds: In every subsequent round r, the adversary initializes p_j^r with 0.0 for all nodes j and then examines all nodes with messages.

If node i is tagged with probability p_i^{r-1} for the previous round and sends k messages in total in the current round, then for every node j to which i sends a number of messages k_j, the adversary adds $\frac{k_j \cdot p_i^{r-1}}{k}$ to p_j^r.

Normalization: At the end of the final round r_{fin} the adversary repeats the same computation as above for the two challenge recipients R_0 and R_1 and then normalizes the probabilities by dividing by each by the sum of the two probabilities.

Guessing: The adversary then guesses that the challenge recipient is the one whose preceding node has the highest probability. If the probabilities are equal or if there is only one node left, then the adversary flips a coin to select a recipient.

Lemma 3 (Perfect forward adversary)**.** *After Normalization, the one-message heuristic forward adversary without output messages, as defined in Definition 7, precisely computed the fraction of instances that have the message end up in each of the final nodes. In particular, the ratio of these fractions corresponds to the privacy loss (cf. Definition 1) for this observation.*

We refer to Appendix 1.3 for the proof.

Theorem 1. *The heuristic adversary as defined in Definition 7 is optimal for recipient anonymity as in Definition 4, i.e., for every adversary A we have*

$$\begin{aligned} &\Pr[b \leftarrow \langle \mathrm{A}|\textsc{Ch}(\Pi, \alpha_{\mathsf{SA}}, 1, b)\rangle \mid b \leftarrow \{0,1\}] \\ \leq\ &\Pr[b \leftarrow \langle \mathrm{A}_{Heu}|\textsc{Ch}(\Pi, \alpha_{\mathsf{SA}}, 1, b)\rangle \mid b \leftarrow \{0,1\}] \end{aligned}$$

Proof. The proof of this theorem follows from what we discussed in this and the previous subsection. We know from Lemma 3 that the heuristic adversary computes the probability that the challenge message ends up with either recipient. The heuristic adversary then guesses the recipient with the highest probability. As any deviation from this guess can only lower the chance of succeeding, the theorem follows. □

5. ε-attack advantage: lower & upper bound

We derive lower and upper bound for the ε-attack advantage $\delta(\varepsilon)$ for any mechanism M by obtaining $\delta(\varepsilon)$ from the privacy loss (cf. Section 3.3). In our work, we first bound an empirical privacy loss distribution (ePLD) $\tilde{\omega}$ (cf. Definition 8) obtained by the attacker from a set of observations O^n with a pointwise confidence band (cf. Definition 9 and Lemma 4) and second use these bounds on the ePLD to derive upper and lower bounds on the ε-attack advantage (cf. Theorem 2). For the second part, we prove that if one PLD is below the other, then so is the respective ε-attack advantage. In Corollary 1, we show that our derived bounds apply to a mixnet using the privacy loss obtained by our optimal attacker (cf. Lemma 3).

5.1. Bounds on the ePLD

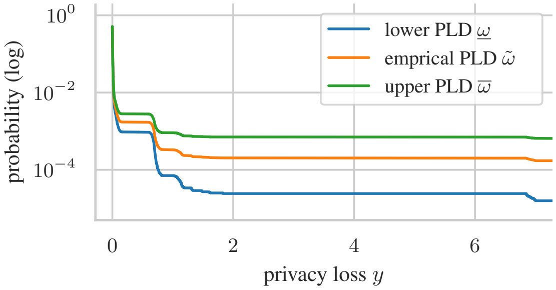


Figure 5: Exemplary cumulative PLDs for $N = 1M, R = 5, k = 1000, |C| = 50$. Leakage continues until $y = \infty$.

We define four privacy loss distributions (PLD) and their respective ε-attack advantages: the actual PLD ω with the ε-attack advantage δ and three reciprocal cumulative PLDs $\underline{\rotatebox[origin=c]{180}{ω}}, \tilde{\omega}, \overline{\rotatebox[origin=c]{180}{ω}}$ with their respective ε-attack advantages $\underline{\delta}, \tilde{\delta}, \overline{\delta}$. Here, $\underline{\rotatebox[origin=c]{180}{ω}}$ is the lower bound PLD with the lower bound ε-attack advantage $\underline{\delta}$, $\tilde{\delta}$ the empirical PLD with the empirical ε-attack advantage $\tilde{\delta}$, and $\overline{\rotatebox[origin=c]{180}{ω}}$ the upper bound LD with the lower and upper bound ε-attack advantage $\overline{\delta}$. With $\rotatebox[origin=c]{180}{ω}$ we notate a reciprocal cumulative PLD to differentiate it from the non-cumulative PLD definition ω. In Figure 5 we visualize the three PLDs $\underline{\rotatebox[origin=c]{180}{ω}}, \tilde{\rotatebox[origin=c]{180}{ω}}, \overline{\rotatebox[origin=c]{180}{ω}}$ for an exemplary mixnet architecture. The exact PLD ω is unknown due to a limited sampling size, but is in Theorem 2 proven to be in between $\underline{\rotatebox[origin=c]{180}{ω}}$ and $\overline{\rotatebox[origin=c]{180}{ω}}$. The empirical PLD $\tilde{\rotatebox[origin=c]{180}{ω}}$ describes the most likely exact PLD ω.

Formally, we define the empirical privacy loss distribution (ePLD) and the exact PLD as follows:

Definition 8 (Empirical privacy loss distribution (ePLD))**.** *Recall from Definition 1 that the PDF of the privacy loss distribution (PLD) for each atomic event $y \in \mathcal{Y}$ is given by*

$$\omega_{M,x_{a:b}}(y) = \sum_{\{o \mid \mathcal{L}_{x_{a:b}}(o)=y, o\in\mathcal{U}\}} \Pr[M(x_a) = o].$$

If we have a finite i.i.d. tuple of observations $O^n := (o_i)_{i=1}^n$ from $\mathcal{U}$, we define the empirical privacy loss distribution (ePLD) with the reciprocal cumulative empirical probability distribution function on support $\mathcal{Y}^q = (y_j)_{j=1}^q = \bigcup_{o\in O^n} \{\mathcal{L}_{x_{0:1}}(o)\} \subseteq \mathcal{Y}$ s.t. $y_j \leq y_{j+1}$ (ascending-sorted), $q \leq n$, $y_0 = 0$, and $y_q = \infty$ as

$$\tilde{\varpi}_{M,x_{a:b},O^n}(y_j) = \frac{1}{n} \underbrace{\sum_{\{o \mid \mathcal{L}_{x_{a:b}}(o)\geq y_j, o\in O^n\}} 1}_{=:k}.$$

Based on ePLD we define the pointwise confidence band.

Definition 9 (Pointwise confidence band of an ePLD)**.** *Given the PLD $\omega_{M,x_{a:b}}$ with support $\mathcal{Y} = (y_i)_{i=1}^m$ and the ePLD $\tilde{\varpi}_{M,x_{a:b},O^n}$ with support $(y_j)_{j=1}^q = \mathcal{Y}^q \subseteq \mathcal{Y}$ (cf. Definitions 1 and 8), we define the ePLD pointwise confidence band $(\tilde{\varpi}_{M,x_{a:b},O^n}, \underline{\varpi}_{M,x_{a:b},O^n}, \overline{\varpi}_{M,x_{a:b},O^n}, \mathcal{Y}^q, \alpha)$ for all $y_j \in \mathcal{Y}^q$ and a nominal coverage probability $1-\alpha$ as*

$$\Pr[\underbrace{\tilde{\varpi}_{M,x_{a:b},O^n}(y_j) - \underline{\beta}_{\alpha}(y_j)}_{=:\underline{\varpi}_{M,x_{a:b},O^n}(y_j)} \leq \sum_{y'\geq y_j} \omega_{M,x_{a:b}}(y')] \geq 1 - \frac{\alpha}{2}$$

$$\Pr[\sum_{y'\geq y_j} \omega_{M,x_{a:b}}(y') \leq \underbrace{\tilde{\varpi}_{M,x_{a:b},O^n}(y_j) + \overline{\beta}_{\alpha}(y_j)}_{=:\overline{\varpi}_{M,x_{a:b},O^n}(y_j)}] \geq 1 - \frac{\alpha}{2}$$

We define for each $y_j \in \mathcal{Y}^q \setminus \{y_q\}$

$$\forall y_i, y_j < y_i < y_{j+1}\colon \underline{\varpi}_{M,x_{a:b},O^n}(y_i) := \underline{\varpi}_{M,x_{a:b},O^n}(y_{j+1})$$
$$\forall y_i, y_j < y_i < y_{j+1}\colon \overline{\varpi}_{M,x_{a:b},O^n}(y_i) := \overline{\varpi}_{M,x_{a:b},O^n}(y_j)$$

which constitutes a step function on the intervals $(y_j, y_{j+1})_j$.

The Clopper-Pearson confidence band resembles an exact implementation of the pointwise confidence band, meaning that the true coverage probability is never less than the required nominal coverage probability [26]. Thus, the true coverage probability might be better than what we report but is impossible to compute given a limited sample size. For an ePLD, the Clopper-Pearson confidence band works as follows:

Lemma 4 (Clopper-Pearson confidence band)**.** *Let the random variable $X^{(n,p)}$ follow the binomial distribution $X^{(n,p)} \sim \mathrm{Bin}(n,p)$ with $k \in \mathbb{N}$ successes in $n \in \mathbb{N}$ independent trails and success rate $p \in [0,1]$. The cumulative distribution function (CDF) of $X^{(n,p)}$ is given by $\Pr[X^{(n,p)} \leq k]$. Then, the ePLD pointwise confidence band $(\tilde{\varpi}_{M,x_{a:b},O^n}, \underline{\varpi}_{M,x_{a:b},O^n}, \overline{\varpi}_{M,x_{a:b},O^n}, \mathcal{Y}^q, \alpha)$ (cf. Definition 9) is for all $y \in \mathcal{Y}^q$ and on coverage probability at least $1-\alpha$*

$$\begin{aligned}&\underline{\varpi}_{M,x_{a:b},O^n}(y)\\ &\quad = \sup_p \{p \mid \Pr[X^{(n,p)} \leq \underbrace{n \cdot \tilde{\varpi}_{M,x_{a:b},O^n}(y_j)}_{=:k}] < \alpha/2\}\\ &\overline{\varpi}_{M,x_{a:b},O^n}(y)\\ &\quad = \inf_p \{p \mid 1 - \Pr[X^{(n,p)} \leq \underbrace{n \cdot \tilde{\varpi}_{M,x_{a:b},O^n}(y_j)}_{=:k}] < \alpha/2\}.\end{aligned}$$

For numerical stability, we can equivalently notate $\underline{\varpi} = B(\alpha/2, k, n-k+1)$ and $\overline{\varpi} = 1 - B(\alpha/2, n-k, k+1)$ with $B(p, \beta_1, \beta_2)$ as the p-th quantile of the beta distribution with shape parameters β_1, β_2.

5.2. Bounds on the ε-attack advantage

Before bounding the ε-attack advantage, we define a partial ordering on PLDs which orders the reciprocal CDFs of two PLDs based on the criterion that the reciprocal CDF of one PLD has to be larger or equal than the reciprocal CDF of the other PLD on all atomic events $y \in \mathcal{Y}$.

Definition 10 (Partial Order on PLDs)**.** *Let ω_1, ω_2 be any PLDs with support $\mathcal{Y}$ as in Definition 1. We say $\omega_1 \leq \omega_2$ iff*

$$\forall y \in \mathcal{Y}\colon\ \textstyle\sum_{y'\geq y} \omega_1(y') \leq \sum_{y'\geq y} \omega_2(y').$$

Due to the properties of the sum, this defines a partial order.

Based on the partial order, we can proof a bucketing relation which shows that if one PLD is smaller than another PLD, then the respective ε-attack advantage is also smaller than the other. We refer to Appendix 1.4 for the proof.

Lemma 5 (Bucketing Relation)**.** *Let ω_1, ω_2 be any PLDs with support $\mathcal{Y}$ as in Definition 1 and δ some ε-attack advantage as in Definition 3. If $\omega_1 \leq \omega_2$ then $\delta(\varepsilon, \omega_1) \leq \delta(\varepsilon, \omega_2)$.*

Based on the bucketing relation, we can prove the upper and lower bound on the ε-attack advantage:

Theorem 2 (Lower and upper bound on the ε-attack advantage δ)**.** *Let $(\tilde{\varpi}_{M,x_{a:b},O^n}, \underline{\varpi}_{M,x_{a:b},O^n}, \overline{\varpi}_{M,x_{a:b},O^n}, \mathcal{Y}^q, \alpha)$ denote the ePLD pointwise confidence band (cf. Definition 9) and $\omega_{M,x_{a:b}}$ a PLD with support $\mathcal{Y} = (y_i)_{i=1}^m$ s.t. $y_i \leq y_{i+1}$ (ascending-sorted) and $(y_j)_{j=1}^q = \mathcal{Y}^q \subseteq \mathcal{Y}$. Then the actual ε-attack advantage $\delta(\varepsilon, \omega_{M,x_{a:b}})$ is bounded with $\underline{\delta}(\varepsilon, \underline{\varpi}_{M,x_{a:b},O^n}) \leq \delta(\varepsilon, \omega_{M,x_{a:b}}) \leq \overline{\delta}(\varepsilon, \overline{\varpi}_{M,x_{a:b},O^n})$ with* *a nominal coverage probability of* $1 - q \cdot \alpha$ *where*

$$\begin{aligned}\underline{\delta}(\varepsilon,\ \underline{\varpi}_{M,x_{a:b},O^n}) &= \underline{\varpi}_{M,x_{a:b},O^n}(\infty)\\ &\quad + \sum_{(y_j)_{j=1}^{q-1}} \max\big(0, (1 - e^{\varepsilon - y_j})\\ &\qquad \cdot (\underline{\varpi}_{M,x_{a:b},O^n}(y_j) - \underline{\varpi}_{M,x_{a:b},O^n}(y_{j+1}))\big)\\ \overline{\delta}(\varepsilon,\ \overline{\varpi}_{M,x_{a:b},O^n}) &= \overline{\varpi}_{M,x_{a:b},O^n}(\infty)\\ &\quad + \sum_{(y_j)_{j=1}^{q-1}} \max\big(0, (1 - e^{\varepsilon - y_{j+1}})\\ &\qquad \cdot (\overline{\varpi}_{M,x_{a:b},O^n}(y_j) - \overline{\varpi}_{M,x_{a:b},O^n}(y_{j+1}))\big).\end{aligned}$$

Proof. We convert the cumulative reciprocal PLDs $\underline{\varpi}, \overline{\varpi}$ to a non-cumulative PLD step function $\underline{\omega}, \overline{\omega}$:

$$\begin{aligned}\forall (y_i)_{i=1}^{m-1} \colon \underline{\omega}_{M,x_{a:b},O^n}(y_i) &= \underline{\varpi}_{M,x_{a:b},O^n}(y_i)\\ &\quad - \underline{\varpi}_{M,x_{a:b},O^n}(y_{i+1})\\ \forall (y_i)_{i=1}^{m-1} \colon \overline{\omega}_{M,x_{a:b},O^n}(y_i) &= \overline{\varpi}_{M,x_{a:b},O^n}(y_i)\\ &\quad - \overline{\varpi}_{M,x_{a:b},O^n}(y_{i+1})\\ \underline{\omega}_{M,x_{a:b},O^n}(y_m) &= \underline{\varpi}_{M,x_{a:b},O^n}(y_m)\\ \overline{\omega}_{M,x_{a:b},O^n}(y_m) &= \overline{\varpi}_{M,x_{a:b},O^n}(y_m).\end{aligned}$$

We write $L \underset{\tilde{\alpha}}{\leq} R$ for an inequality that holds with a coverage probability of $1 - \tilde{\alpha}$: $\Pr[L \leq R] \geq 1 - \tilde{\alpha}$. By the Bonferroni correction, it suffices to show for a coverage probability of $1 - \tilde{\alpha}$ that two inequalities each hold with coverage probability $1 - \tilde{\alpha}/2$. The Bonferroni correction [15] follows from the union bound.

We now prove this theorem as follows: in *Case 1* we show that $\underline{\delta}(\varepsilon, \underline{\omega}_{M,x_{a:b},O^n}) \underset{q\,\cdot\,\alpha/2}{\leq} \delta(\varepsilon, \omega_{M,x_{a:b}})$ and in *Case 2* we show $\delta(\varepsilon, \omega_{M,x_{a:b}}) \underset{q\,\cdot\,\alpha/2}{\leq} \overline{\delta}(\varepsilon, \overline{\omega}_{M,x_{a:b},O^n})$. By Lemma 5, it suffices to show that (*Case 1*) $\underline{\omega}_{M,x_{a:b},O^n} \underset{q\,\cdot\,\alpha/2}{\leq} \omega_{M,x_{a:b}}$ and (*Case 2*) $\omega_{M,x_{a:b}} \underset{q\,\cdot\,\alpha/2}{\leq} \overline{\omega}_{M,x_{a:b},O^n}$.

Case 1. We show $\forall y_i \in \mathcal{Y} \colon \underline{\omega}_{M,x_{a:b},O^n} \underset{q\,\cdot\,\alpha/2}{\leq} \omega_{M,x_{a:b}}$. W.l.o.g. we assume that $y_j < y_i \leq y_{j+1}$ for all $y_j \in \mathcal{Y}^q \setminus \{y_q\}$. By Definition 10 and the definition of $\underline{\omega}$, we have

$$\underline{\omega}_{M,x_{a:b},O^n}(y_i) = \sum_{y' \geq y_i} \underline{\omega}_{M,x_{a:b},O^n}(y') = \underline{\varpi}_{M,x_{a:b},O^n}(y_i)$$

and by the step-function continuation of $\underline{\varpi}$ in Definition 9

$$= \underline{\varpi}_{M,x_{a:b},O^n}(y_{j+1}).$$

For the other side of the inequality, we have

$$\begin{aligned}\omega_{M,x_{a:b}}(y_i) &= \sum_{y' \geq y_i} \omega_{M,x_{a:b}}(y')\\ &= \sum_{y' \geq y_{j+1}} \omega_{M,x_{a:b}}(y') + \sum_{y_{j+1} > y' \geq y_i} \omega_{M,x_{a:b}}(y')\\ &\geq \sum_{y' \geq y_{j+1}} \omega_{M,x_{a:b}}(y').\end{aligned}$$

Thus, putting both sides of the inequality together, we have

$$\underline{\varpi}_{M,x_{a:b},O^n}(y_{j+1}) \underset{q\,\cdot\,\alpha/2}{\leq} \sum_{y' \geq y_{j+1}} \omega_{M,x_{a:b}}(y') \tag{1}$$

which holds with coverage probability $1 - \alpha/2$ by Definition 9 for a given y_j. The point y_0 is not contained in any interval. Yet, by the same argumentation where we have $y_i = y_0$, we conclude $\underline{\varpi}_{M,x_{a:b},O^n}(y_0) \underset{q\,\cdot\,\alpha/2}{\leq} \sum_{y' \geq y_0} \omega_{M,x_{a:b}}(y')$. For all $y_j \in \mathcal{Y}^q$, Inequality 1 also holds by the Bonferroni correction since we use a coverage probability of $1 - q \cdot \alpha/2$.

We simplify the calculation of $\underline{\delta}(\varepsilon, \underline{\omega}_{M,x_{a:b},O^n})$ using the step-function continuation, i.e. $\underline{\varpi}_{M,x_{a:b},O^n}(y_i) = \underline{\varpi}_{M,x_{a:b},O^n}(y_{j+1})$ in the interval $y_j < y_i \leq y_{j+1}$. Thus, we only have to sum over $\mathcal{Y}^q$ instead of $\mathcal{Y}$:

$$\begin{aligned}&\underline{\delta}(\varepsilon, \underline{\varpi}_{M,x_{a:b},O^n})\\ &= \underline{\varpi}_{M,x_{a:b},O^n}(\infty) + \sum_{(y_i)_{i=1}^{m-1}} \max\big(0, (1 - e^{\varepsilon - y_i})\\ &\qquad \cdot (\underline{\varpi}_{M,x_{a:b},O^n}(y_i) - \underline{\varpi}_{M,x_{a:b},O^n}(y_{i+1}))\big)\\ &= \underline{\varpi}_{M,x_{a:b},O^n}(\infty) + \sum_{(y_j)_{j=1}^{q-1}} \max\big(0, (1 - e^{\varepsilon - y_j})\\ &\qquad \cdot (\underline{\varpi}_{M,x_{a:b},O^n}(y_j) - \underline{\varpi}_{M,x_{a:b},O^n}(y_{j+1}))\big)\end{aligned}$$

Case 2 follows similarly; we refer to Appendix 1.5 for the detailed proof. □

If we can not assume a symmetric privacy loss, i.e. $\omega_{M,x_{0:1}} = \omega_{M,x_{1:0}}$, then we have to use the ε-attack advantage $\delta'(\varepsilon, \omega_{M,x_{0:1}}, \omega_{M,x_{1:0}}) = \max_{(a,b) \in \{(0,1);(1,0)\}} \delta(\varepsilon, \omega_{M,x_{a:b}})$ (cf. Definition 3) and derive from Theorem 2:

$$\begin{aligned}\max_{\substack{(a,b)\in\\ \{(0,1);(1,0)\}}} \underline{\delta}(\varepsilon, \overline{\varpi}_{M,x_{a:b},O^n}) &\leq \delta(\varepsilon, \omega_{M,x_{a:b}})\\ &\leq \overline{\delta}(\varepsilon, \underline{\varpi}_{M,x_{a:b},O^n}).\end{aligned}$$

In this case, the coverage probability also changes to $1 - (q_{0:1} + q_{1:0}) \cdot \alpha/2$ where $q_{0:1}$ is the size of the support of the ePLD $\tilde{\varpi}_{M,x_{0:1},O^n}$ and $q_{1:0}$ is the size of the support of the ePLD $\tilde{\varpi}_{M,x_{1:0},O^n}$.

5.3. Bounding the ε-attack advantage of mixnets

From Lemma 3, we know that for every observation the optimal attacker outputs the precise privacy loss. Hence, we can conclude that the following two probabilistic experiments are equally distributed for all privacy losses y.

- *Experiment 1*: Sample a privacy loss $\hat{y}$ from the real PLD of a given mixnet μ. If $\hat{y} < y$, output 0; otherwise, output 1.
- *Experiment 2*: Sample a randomness tape r; compute an observation from the mixnet μ from r; run the heuristic adversary and compute a privacy loss $\hat{y}$. If $\hat{y} < y$, output 0; otherwise, output 1.

For any input, we can get a pair of output distributions (the metadata observations) from the AnoA recipient anonymity game for a mixnet μ, one distribution for $b = 0$ and one for $b = 1$. So, for the experiment that independently samples from experiment 2, we conclude from Lemma 4 and Theorem 2 a lower and an upper point for the ε-attack advantage for each pair of observation distributions from the AnoA game for μ. Since any pair of distinct recipients $\mathsf{R}_1, \mathsf{R}_2$ constitute worst-case inputs for the recipient anonymity game with mixnet μ, the upper and lower bound on the ε-attack advantage hold for any attacker.

Corollary 1. *Given a mixnet μ and a pair of recipients $\mathsf{R}_1, \mathsf{R}_2$, when independently running Experiment 2 from above q times, $\underline{\delta}(\varepsilon, \underline{\omega}_{\mu,\mathsf{R}_{a:b},O^n})$ and $\overline{\delta}(\varepsilon, \overline{\omega}_{\mu,\mathsf{R}_{a:b},O^n})$ are a lower bound and an upper bound, respectively, for the ε-attacker advantage $\delta(\varepsilon, \omega_{\mu,\mathsf{R}_{a:b}})$ for recipient anonymity with a coverage probability of $1 - q \cdot \alpha$.*

6. Evaluation

6.1. Implementation

We implemented the heuristic adversary in Python as follows. A single instance of the heuristic adversary starts by selecting the challenge bit b determining the challenge recipient for the experiment. We simulate one run of a stratified network, where each user sends exactly one message. For each message, we determine independently, uniformly at random the path that this message will take.

We implement three recipients: two challenge recipients R_0 and R_1, and a noise recipient $\mathsf{R}_{\text{noise}}$. Our challenge user u_0 sends a message based on the challenge bit b to the recipient R_b. All other users, except for one that is chosen uniformly at random, send their message to $\mathsf{R}_{\text{noise}}$. The chosen random user sends a message to R_{1-b} instead.

In our script, we compute and track the probability distribution from the optimal adversary's point of view of where the challenge message is. This probability is spread over nodes as well as over messages:

- Probability on nodes means that the adversary assigns a certain probability to the challenge message being in a given node. This is done for honest nodes only, as the adversary has deeper insights into the inner workings of compromised nodes. Messages leaving an honest node each get a share of the probability from the node and carry it to the subsequent node (if that is also an honest node).
- Probability on messages means that the adversary is tracking specific messages and the probability that they are the challenge message. This is done both at the start of the protocol (where the challenge message is known) and whenever messages traverse compromised nodes. In those instances, they ”carry some of the probability from the previous node” and keep it attached to the message until they reach an honest node again.

Our script then proceeds as follows, computing the probabilities from the point of view of the optimal adversary: for the first round, all nodes are assigned a probability of 0.0. The challenge message itself is assigned a probability of 1.0 (we initially know that it is the challenge message).

For each node in each round, we track the probability that the optimal adversary assigns to that node and the number of messages in that node at that round. We let every message carry the probability from the previous round to the next round as follows: if the node is honest, we add the probability to said node. If the node is dishonest, we keep it on the message itself (the adversary keeps its beliefs about the message). For every round, message, and packet we then proceed as follows:

1) We consider some of the nodes of each round to be compromised. For ease of implementation, we sort the nodes per layer so that the compromised nodes are on top. As all choices are performed uniformly at random, this sorting does not impact our calculations.

2) We compute for each message and node the propagation of probabilities as follows.

3) If a message carries a probability and this message is routed through a compromised node, it retains said probability and does not affect anything else. The adversary can trace this message and will retain its beliefs about the message.

4) If a message carries a probability and enters an honest node, it transfers the probability to that node.

5) If a message leaves an honest node, it carries a part of the total probability in that node. If the node has a total probability of p and k messages are inside, each carries $^{p}\!/_{k}$.

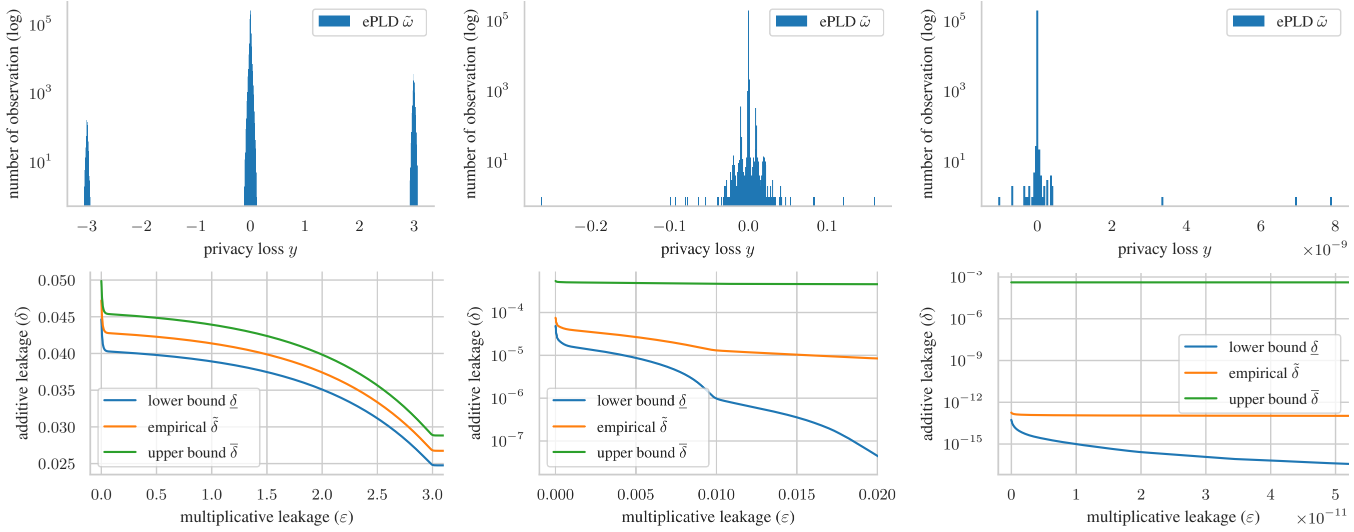
Eventually, by Lemma 3 we get the privacy loss by the ratio of the probabilities assigned to R_b and R_{1-1} at the end of the run. We then output this privacy loss.

This process assigns to each node the probability of holding the challenge message. We carry out the calculation until the messages reach their recipients ($\mathsf{R}_0, \mathsf{R}_1, \mathsf{R}_{\text{noise}}$). The challenge message, as well as the message from the one uniformly chosen other user, reach the two challenge recipients R_0 and R_1. All remaining messages reach a special noise recipient $\mathsf{R}_{\text{noise}}$. As the adversary knows by the definition of the game that the challenge message is never sent to $\mathsf{R}_{\text{noise}}$, they can ignore those and normalize their probabilities p_0 and p_1 for R_0 and R_1 respectively.

For this observation, we can then compute the privacy loss y for this run as $\ln \frac{p_b}{p_{1-b}}$. If $p_{1-b} = 0$, we set $y = 10$ and treated all $y = 10$ losses as $y = \infty$ in our privacy leakage evaluation. By running the script for many iterations we get a list of observed privacy losses. Each of those is generated from basic uniformly distributed and independent coin tosses, i.e., the underlying events generating them have the same probability, which allows us to treat all observed privacy losses as equally probable and thus to construct a privacy loss distribution from them.

6.2. GadgetTester

We also implemented a way to test gadgets such as those used for Karaoke. In this instance, we generate observations as above and compute the privacy loss. Additionally, we check for each observation whether the gadget conditions



(a) $R = 3, k = 20, |C| = 2$ (b) $R = 5, k = 100, |C| = 2$ (c) $R = 15, k = 20, |C| = 1$

Figure 6: Empirical privacy loss distribution (ePLD) without distinguishing events (top) and ε-attack advantage $\delta(\varepsilon)$ (bottom) of three mixnet architectures with the multiplicative leakage ε and additive one δ. We plot the lower $\underline{\delta}(\varepsilon)$ and upper bound $\overline{\delta}(\varepsilon)$ of $\delta(\varepsilon)$ and its empirical estimate $\tilde{\delta}(\varepsilon)$. Each lower and upper bound holds with a coverage probability of $1 - 10^{-30}$.

(c.f. our discussion of Karaoke in Section 2) are satisfied. If they are, we keep the observation, otherwise we discard it. We can then report on the observation that satisfied the gadget but has the highest observed privacy loss. This can be used to quickly generate good counter-examples against the intuition that the gadget leads to an observation where the adversary cannot win.

6.3. Experiments

We evaluate the heuristic (optimal) adversary for a range of networks with different user numbers N, lengths R (how many rounds does a node stay in the network), widths k (how many nodes per round are there to choose from per round), compromisation $|C|$ (how many nodes per layer are compromised). For all those scenarios we compute a range of measures by first generating many iterations of runs of our adversary through the network (c.f., our implementation in Section 6.1). This way we end up with a large number of sampled privacy loss values. As each of our choices that lead to the computation of the privacy loss follows a uniform distribution, we can combine our observed privacy loss values to form an empirical approximation of the privacy loss distribution. Based on this distribution we then compute:

- The expected privacy loss of worst-case recipients on a set of observations O^n without distinguishing events: $E_{O^n}[y]$. This is the same as the PLD's first moment, which in the literature is also described as $\rho(\alpha = 1)$ in Rényi Differential Privacy (RDP) [25].
- The empirical ε-attack advantage $\tilde{\delta}(\varepsilon)$.
- A lower bound $\underline{\delta}(\varepsilon)$ and an upper bound $\overline{\delta}(\varepsilon)$ for the ε-attack advantage. Each holds with a coverage probability of $1 - 10^{-30}$.

Concretely, we evaluate a range of settings, starting from just $R = 3$ rounds, up to 15 rounds with a network width of $k = 20, 100$, or even 1000 and for a variety of fractions of compromised nodes. We mostly evaluate the $N = 1M$ user setting but also 100 users for a range of small networks.

We find that, naturally, anonymity improves significantly with the number of rounds of mixing and decreases significantly with the width of the network, as either of those directly impacts the probability of packets meeting each other and mixing. In observing the privacy loss values, we clearly see that even for generally great networks, the privacy loss is very often not exactly zero, even if packets generally mix well. We report cases where the adversary outright or almost surely wins as *distinguishing events* $\delta(\infty)$. For most network parameters (5+ rounds and not too many nodes) those occur very rarely as they tend to be the result of packets traveling on very isolated paths or through compromised nodes.

We present our results in Table 2 and observe that networks with just $R = 3$ rounds and a significant width are not resilient against even 5% compromisation. The adversarial advantage is so high that even our lower bound reaches 10-20% for networks of width $k = 1000$, and even the chance for total anonymity failure is quite significant. These results confirm existing work in this area. Novel, however, is the realization that even if the probability for a distinguishing event is small, the adversarial advantage can still be notably higher. Even networks with few rounds, the empirical estimate for the advantage and even the lower bound on $\delta(0)$ is typically higher than even the upper bound on $\delta(\infty)$. Most notably, for $R = 10+$ rounds we did not observe even a single distinguishing event, but many of the observations still had a notable privacy loss as reflected in our estimate and bounds for $\delta(0)$.

TABLE 2: Selected privacy leakages of different mixnet architectures. We report 1) the expected empirical privacy loss or ε on a set of observations O^n without distinguishing events, $E_{O^n}[y]$, and its standard error σ^-; negative values are due to the sampling error, 2) the lower bound $\underline{\delta}(\infty)$ and upper bound $\overline{\delta}(\infty)$ of the probability on distinguishing events $\delta(\infty)$ and its empirical estimate $\tilde{\delta}(\infty)$, and 3) the lower bound $\underline{\delta}(0)$ and upper bound $\overline{\delta}(0)$ of the attack advantage $\delta(0)$ and its empirical estimate $\tilde{\delta}(0)$. Each lower and upper bound holds with a coverage probability of $1 - 10^{-30}$.

MixNet Architecture					Privacy Leakage		
Name	N	R	k	$\|C\|$	$E_{O^n}[y] \pm \sigma^-$	$\underline{\delta}(\infty) \leq \tilde{\delta}(\infty) \leq \overline{\delta}(\infty)$	$\underline{\delta}(0) \leq \tilde{\delta}(\mathit{0}) \leq \overline{\delta}(0)$
–	100	3	20	1	0.4 ± 1e-3	0.2 ≤ *0.3* ≤ 0.3	0.5 ≤ *0.5* ≤ 0.5
–	100	3	20	2	0.4 ± 1e-3	0.3 ≤ *0.3* ≤ 0.3	0.5 ≤ *0.5* ≤ 0.6
–	100	5	5	1	0.02 ± 4e-4	4e-3 ≤ *6e-3* ≤ 8e-3	0.04 ≤ *0.05* ≤ 0.05
–	100	5	20	1	0.06 ± 3e-4	2e-3 ≤ *3e-3* ≤ 4e-3	0.1 ≤ *0.1* ≤ 0.1
–	100	5	20	2	0.1 ± 5e-4	0.01 ≤ *0.01* ≤ 0.01	0.2 ≤ *0.2* ≤ 0.2
–	1M	3	20	1	0.01 ± 2e-4	6e-3 ≤ *7e-3* ≤ 8e-3	0.01 ≤ *0.01* ≤ 0.02
–	1M	3	20	2	0.05 ± 4e-4	0.02 ≤ *0.03* ≤ 0.03	0.04 ≤ *0.05* ≤ 0.05
–	1M	3	1000	50	0.08 ± 6e-4	0.04 ≤ *0.04* ≤ 0.04	0.1 ≤ *0.1* ≤ 0.1
–	1M	3	1000	100	0.2 ± 1e-3	0.08 ≤ *0.09* ≤ 0.09	0.2 ≤ *0.2* ≤ 0.2
–	1M	5	20	1	8e-5 ± 2e-5	3e-7 ≤ *2e-5* ≤ 1e-4	2e-4 ≤ *2e-4* ≤ 4e-4
–	1M	5	20	2	1e-3 ± 6e-5	2e-4 ≤ *4e-4* ≤ 7e-4	1e-3 ≤ *1e-3* ≤ 2e-3
–	1M	5	20	4	0.01 ± 5e-4	5e-3 ≤ *6e-3* ≤ 9e-3	0.01 ≤ *0.01* ≤ 0.02
Atom	1M	5	100	2	2e-6 ± 3e-6	3e-36 ≤ *5e-6* ≤ 4e-4	5e-5 ≤ *8e-5* ≤ 5e-4
–	1M	5	1000	50	8e-4 ± 9e-5	2e-5 ≤ *2e-4* ≤ 6e-4	3e-3 ≤ *3e-3* ≤ 4e-3
–	1M	5	1000	100	6e-3 ± 3e-4	9e-4 ≤ *2e-3* ≤ 3e-3	0.01 ≤ *0.01* ≤ 0.02
–	1M	10	20	1	-2e-9 ± 2e-9	0 ≤ *0* ≤ 3e-4	3e-9 ≤ *4e-9* ≤ 4e-4
Atom	1M	10	100	2	-1e-11 ± 3e-11	0 ≤ *0* ≤ 5e-4	2e-10 ≤ *2e-10* ≤ 7e-4
Karaoke	1M	10	1000	200	6e-5 ± 5e-5	0 ≤ *0* ≤ 1e-3	2e-4 ≤ *4e-4* ≤ 2e-3
Karaoke	1M	14	20	4	-4e-8 ± 2e-8	0 ≤ *0* ≤ 3e-4	9e-8 ≤ *2e-7* ≤ 4e-4
Karaoke	1M	14	1000	200	-6e-7 ± 7e-7	0 ≤ *0* ≤ 3e-3	2e-6 ≤ *4e-6* ≤ 4e-3
–	1M	15	20	1	9e-14 ± 6e-14	0 ≤ *0* ≤ 3e-4	5e-14 ≤ *2e-13* ≤ 4e-4
Atom	1M	15	100	2	6e-17 ± 4e-17	0 ≤ *0* ≤ 6e-4	5e-16 ≤ *7e-16* ≤ 7e-4

In Figure 6 we present this leakage of mixnets even more directly inspired by approximate differential privacy (ADP): for a given privacy parameter ε we present the value $\delta(\varepsilon)$ we need so that the protocol is (ε, δ)-differentially private. Here, we observe a graceful decline of the ε-attack advantage between the attack advantage $\delta(0)$ and the probability of distinguishing events $\delta(\infty)$. We also plot the ePLD (cf. Definition 8) which shows the distribution of privacy leakage among all observations. All of these have a large spike at a zero privacy loss in common, yet all follow different distributions for non-zero losses.

7. Implications and discussion

Our heuristic adversary and its experimental evaluations have several impacts:

1) Our heuristic adversary by its very existence demonstrates that attackers can efficiently leverage the biases that stem from unequal network edge saturation.

2) Our methodology can be adapted to thoroughly evaluate any new protocol or existing protocol, or to verify the *gadget* used in their security proofs.

3) We draw important insights about the anonymity guarantees of Karaoke (and its successors), Atom-like networks, and guidance towards designing future protocols.

Now we present the specific insights from our experimental evaluations about different mixnet-based protocols.

7.1. Karaoke and its successors

Based on our implementation of the gadget tester in Section 6.2, we evaluate the *gadget* used in the anonymity proof of Karaoke. Our gadget tester demonstrates the inaccuracy of their proof by producing counter-examples that break anonymity but satisfy the conditions of their gadget. Despite the flawed security proof, we want to understand what kind of guarantees Karaoke-like systems can provide for different configurations (number of rounds, total number of messages, percentage of compromised nodes).

If we consider 20% compromised mixnodes in Karaoke, the adversarial advantage $\tilde{\delta}(0)$ is estimated to be 2e-7 or 4e-6 (depending on the width of the network), which could be considered low enough. However, we note that the lower bound on the adversarial advantage is quite similar to our estimate, at 9e-8 and 2e-6 respectively.

The privacy loss slightly increases with the width of the network, however, it does not increase proportionally: which implies that increasing the width of the network for the benefit of scalability does not drastically destroy anonymity.

These insights can be directly translated to the anonymity guarantees of Yodel [23] and Groove [5]. They achieve unlinkability at the circuit-level (which user is talking to which other user), and not at the packet-level: each packet corresponding to the same circuit would still be linkable. However, that is expected in their application scenarios like voice calls.

7.2. Possible relaxations for Atom

Even though the square shuffle technique achieves a close-to-uniform shuffle in less than 10 rounds, it requires all the mixers to be honest. Because of that, Atom needs to use a cluster of servers. Each such cluster consists of 32 servers in an anytrust model, to emulate such honest mixers. That introduces a significant overhead on Atom in terms of latency. Some natural questions are: (1) Can we use smaller clusters and still achieve strong guarantees? (2) Would that be strictly better than using mixnets that do not employ such clusters? (3) Do we still need exactly the same number of messages on each link?

Our results show that we can actually relax those restrictions. Our experiments use one million messages and 100 mixers for every hop/round, and the path of each message is chosen independently of all other messages. Even if each mixer has only 4 servers, and a total of 20% of the network is compromised, the probability that each mixer is compromised is less than 2%. In such a scenario, the network achieves strong anonymity guarantees with 10 rounds (the number of recommended rounds also in the original Atom [20] paper). However, the total number of servers traveled by a packet is reduced by $1/8$, which is a significant improvement.

If Atom has only 5 rounds (with 4 servers in each mixing cluster), still the anonymity degree we observed is significantly stronger than that of Karaoke with 15 rounds. With a naïve deployment, each message in Atom will travel through 20 servers on their path, in contrast to only 15 in Karaoke. A clever deployment could further optimize the communication among servers inside a cluster (especially in a round-based communication model), and could significantly improve the end-to-end latency. This invites the research community to investigate techniques for such possible optimizations for Atom-like networks.

7.3. General insights about mixnets

We draw the following general insights about mixnets: mixnets have substantial leakage left out of prior analyses that stems from small biases in probabilities of making the same observation of network traffic in different scenarios. A clever adversary can easily and efficiently leverage this leakage just by looking at the density of network traffic. Such an adversary can often be about one order of magnitude stronger than one that relies on total anonymity failure. Methods for measuring and discussing differential privacy can be an interesting tool for further study of mixnet designs as they allow to distinguish between small biases (ε) and possibly catastrophic events (δ).[3]

Additionally, our evaluation confirms some of the conjectures related to mixnet designs: 1) It is well known that anonymity is difficult to achieve with low latency. Our experiments confirm that leakage is high when the protocol has only 3 or 5 rounds (except Atom, which has multiple servers in each cluster). Naturally, the leakage we observe for these networks is significantly higher than the proven lower bounds of the anonymity trilemma of Das et al. [10], [12]. 2) When there are 10 or more rounds and many users ($N = 1M$), the probability of total distinguishing events reduces significantly, to the point where we did not observe a single such event in our evaluation. Overall privacy leakage also reduces, but depending on other parameters can remain an issue. 3) Our evaluations show slow degradation in privacy with the width of the network. For most existing provable mixnet designs, the derived guarantees degrade almost linearly with the width of the network. This provides evidence of untightness in the proof techniques when there are many honest messages in the system, and possibly those proof techniques could be improved to derive tighter guarantees.

7.4. Interdependent messages

Our adversary uses a very precise and helpful heuristic to track individual messages. This heuristic is, as we have shown, optimal for the anonymity notions we consider. However, there is a notable limitation to the optimality of our approach: cases in which tracking multiple interdependent messages helps.

Example. Consider a relationship anonymity game that slightly differs from the one-message relationship anonymity game as presented in AnoA: There are two senders, Alice and Bob, and two recipients, Charlie and Dave. Alice and Bob both send a message and the goal of the adversary is to determine whether Alice's message is sent to Charlie and Bob's message to Dave, or whether Alice's message is sent to Dave and Bob's message to Charlie. In such a game, the adversary can not only win by determining, say, where Alice's message goes, but also by determining where Bob's message doesn't go. If an observation only allows for either Alice's message to go to Charlie OR Bob's message to go to Dave, but not both at the same time (e.g., there is an edge with weight 1 that would have to be taken by both messages at the same time), then the adversary can exclude one of the challenge bits. A weaker version of this interference has the probabilities of Bob's message's path depend on which path Alice's message has taken. In such cases, our adversary is not optimal and its success only presents a lower bound on the adversarial success.

7.5. Limitations

Our adversary is designed specifically for mixnet-type systems, and cannot be used for MPC-based protocols [9], [16] or DC-nets [4], [11], [17]. Moreover, we prove the optimality of our adversary only for recipient anonymity in this paper; the same adversary is not optimal for notions like relationship anonymity (where information about more than one message can be leveraged), though it will translate to sender anonymity if it is cast as a mirror of recipient anonymity. Moreover, our adversary might not be optimal for other notions (e.g., entropy-based anonymity notion).

[3] This δ also captures probability mass outside of the ε bound.

8. Conclusion

In this paper, we have shown that mixnets exhibit a hitherto unexplored dimension of leakage that stems from small (but relevant) biases in a posteriori probabilities of observations. We have described these biases using the privacy loss from the differential privacy literature and have described an adversary that leverages any such biases. Our adversary is efficient and provably optimal for recipient anonymity and could be expanded to other notions as the field explores this leakage in the future. Based on this optimal adversary, we have also performed extensive empirical evaluations of mixnet structures to examine the impact of the number of rounds, the width of the network, and the degree of compromisation on the privacy loss and consequently on the adversarial advantage. Taking a posteriori biases into account leads to far tighter lower bounds (often by about one order of magnitude) than just quantifying the probability of a total breakdown of anonymity. Our lower bounds provide guidance away from leaky mixnet designs, e.g., using too few rounds, and our empirical estimates provide a good overview of can be expected.

References

[1] 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

[2] Ö. Askin, T. Kutta, and H. Dette, "Statistical quantification of differential privacy: A local approach," in *43rd IEEE Symposium on Security and Privacy (S& P '22)*. IEEE, 2022, pp. 402–421.

[3] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi, "Anoa: A framework for analyzing anonymous communication protocols," *Journal of Privacy and Confidentiality*, vol. 7, no. 2, 2016.

[4] L. Barman, I. Dacosta, M. Zamani, E. Zhai, A. Pyrgelis, B. Ford, J. Feigenbaum, and J.-P. Hubaux, "Prifi: Low-latency anonymity for organizational networks," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, pp. 24–47, 10 2020.

[5] L. Barman, M. Kol, D. Lazar, Y. Gilad, and N. Zeldovich, "Groove: Flexible Metadata-Private messaging," in *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*. Carlsbad, CA: USENIX Association, Jul. 2022, pp. 735–750. [Online]. Available: https://www.usenix.org/conference/osdi22/presentation/barman

[6] B. Bichsel, T. Gehr, D. Drachsler-Cohen, P. Tsankov, and M. T. Vechev, "Dp-finder: Finding differential privacy violations by sampling and optimization," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*, D. Lie, M. Mannan, M. Backes, and X. Wang, Eds. ACM, 2018, pp. 508–524.

[7] B. Bichsel, S. Steffen, I. Bogunovic, and M. T. Vechev, "Dp-sniper: Black-box discovery of differential privacy violations using classifiers," in *42nd IEEE Symposium on Security and Privacy (S& P '21)*. IEEE, 2021, pp. 391–409.

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

[9] H. Corrigan-Gibbs, D. Boneh, and D. Mazières, "Riposte: An anonymous messaging system handling millions of users," in *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 321–338.

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

[11] D. Das, E. Mangipudi, and A. Kate, "Organ: Organizational anonymity with low latency," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, pp. 582–605, 07 2022.

[12] D. Das, S. Meiser, E. Mohammadi, and A. Kate, "Comprehensive anonymity trilemma: User coordination is not enough," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, pp. 356–383, 07 2020.

[13] ——, "Divide and funnel: A scaling technique for mix-networks," in *2024 IEEE 37th Computer Security Foundations Symposium (CSF)*, 2024, pp. 49–64.

[14] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies*, ser. PET'02. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 54–68.

[15] O. J. Dunn, "Multiple comparisons among means," *Journal of the American Statistical Association*, vol. 56, pp. 52–64, 1961.

[16] S. Eskandarian, H. Corrigan-Gibbs, M. Zaharia, and D. Boneh, "Express: Lowering the cost of metadata-hiding communication with cryptographic privacy," *ArXiv*, vol. abs/1911.09215, 2019.

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

[18] I. B. Guirat, D. Gosain, and C. Diaz, "Mixim: Mixnet design decisions and empirical evaluation," in *WPES '21: Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society, Virtual Event, Korea, 15 November 2021*. Virtual Event, Korea: ACM, 2021, pp. 33–37. [Online]. Available: https://doi.org/10.1145/3463676.3485613

[19] J. Håstad, "The square lattice shuffle," *Random Structures and Algorithms*, vol. 29, no. 4, p. 466–474, Dec. 2006.

[20] A. Kwon, H. Corrigan-Gibbs, S. Devadas, and B. Ford, "Atom: Horizontally scaling strong anonymity," in *Proceedings of the 26th Symposium on Operating Systems Principles*, ser. SOSP '17, 2017, p. 406–422.

[21] A. Kwon, D. Lu, and S. Devadas, "Xrd: Scalable messaging system with cryptographic privacy," in *Symposium on Networked Systems Design and Implementation*, 2020.

[22] D. Lazar, Y. Gilad, and N. Zeldovich, "Karaoke: Distributed private messaging immune to passive traffic analysis," in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, Carlsbad, CA, Oct. 2018, pp. 711–725.

[23] ——, "Yodel: Strong metadata security for voice calls," in *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, ser. SOSP '19, 2019, p. 211–224.

[24] S. Meiser and E. Mohammadi, "Tight on Budget? Tight Bounds for r-Fold Approximate Differential Privacy," in *Proceedings of the 25th ACM Conference on Computer and Communications Security (CCS)*. ACM, 2018.

[25] I. Mironov, "Rényi differential privacy," *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, 2017.

[26] R. G. Newcombe, "Two-sided confidence intervals for the single proportion: comparison of seven methods." *Statistics in medicine*, vol. 17 8, pp. 857–72, 1998.

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

[28] D. Sommer, S. Meiser, and E. Mohammadi, "Privacy loss classes: The central limit theorem in differential privacy," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 2, pp. 245–269, 2019.

[29] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, and N. Zeldovich, "Stadium: A distributed metadata-private messaging system," 10 2017, pp. 423–440.

Appendix

1. Postponed proofs

1.1. Proof for Lemma 1.

Proof. Given that in round r a node i has n messages in it and edges $w_1, \ldots, w_k$, we compute how many many ways there are to distribute those n messages over the k nodes to which there are edges. This computation boils down to a rather straight-forward mathematical calculation.

Looking at any one edge w_j, then given the partial information about its weight and about which n messages are in node i, there are $\binom{n}{w_j} = \frac{n!}{w_j! \cdot (n-w_j)!}$ many combinations of messages we could select. For the next node, we then have $n' = n - w_j$ messages left to choose from.

For any arbitrary ordering of edges, starting from the first edge and iterating over them all, keeping in mind that $n = \sum_{j=1}^{k} w_j$, the general form is that there are

$$\frac{n! \cdot \prod_{j=1}^{k-1}(n - \sum_{h=1}^{j} w_h)!}{\prod_{j=1}^{k} w_j! \cdot \prod_{j=1}^{k}(n - \sum_{h=1}^{j} w_h)!} = \frac{n!}{\prod_{j=1}^{k} w_j! \cdot 1}$$

combinations in total and hence that many partial instances consistent with the information about the node. □

1.2. Proof for Lemma 2.

Proof. Let $w = [w_1, \dots, w_k]$ denote the edge weights, where k is the number of nodes in the subsequent layer. Note that from Lemma 1 we know that the total number of partial instances compatible with information about this one node is $\frac{\sum_{j=1}^{k} w_j!}{\prod_{j=1}^{k} w_j!} = \frac{n!}{\prod_{j=1}^{k} w_j!}$. If the message m goes to node x with edge weight from i to x being w_x, the following two things will happen:

1) The adjusted edge weight w' (of all remaining messages) going to node x decreases by 1, i.e., $w'_x = w_x - 1$
2) Naturally, the sum of adjusted edge weights is $\sum_{j=1}^{k} w'_j = \sum_{j=1}^{k} w_j - 1 = n - 1$.

By Lemma 1 we then compute the number of instances that have message m travel to node x as:

$$\frac{(\sum_j w'_j)!}{\prod_j w'_j!} = \frac{(\sum_j w'_j)!}{\prod_{j \neq x} w_j! \cdot w_x!} = \frac{(\sum_j w'_j)!}{\prod_j w_j! \cdot \frac{1}{w_x}}$$
$$= w_x \cdot \frac{(\sum_j w'_j)!}{\prod_j w_j!} = w_x \cdot \frac{(n-1)!}{\prod_j w_j!}$$

We see that the number of instances has a common factor of $\frac{(n-1)!}{\prod_j w_j!}$ for every choice of node x. The only impact of choosing a node x is the factor w_x. Since the total number of partial instances was $\frac{n!}{\prod_{j=1}^{k} w_j!}$ it directly follows that a fraction of $\frac{w_x}{n}$ of those has message m in node x. □

1.3. Proof for Lemma 3.

Proof. We show this by induction.

Induction start: The message is in a known node in one layer. We apply Lemma 2 and get that for each of the edge weights, the probability that the message is in the corresponding node in the next layer is a fraction which is determined by the edge weight. That's precisely what the forward adversary outputs.

Induction step: Assume that we have a (correctly computed) distribution of message probabilities over the nodes in a layer. More precisely, for every node i in the current layer, we have that a fraction of instances f_i has the message in that node. By Lemma 2 we can compute a fraction $f_{i,j}$ of each of those instances that will subsequently place the message in node j of the next layer if it was in node i previously. The total fraction of instances that place a message in a node j then is given by $f_j = \sum_i (f_i \cdot f_{i,j})$, which is precisely what the forward adversary computes. □

1.4. Proof for Lemma 5.

Proof. First we observe that for all $0 \leq c \leq d$, and all $y_i, y_j \in \mathcal{Y}$ with $i \leq j$ we have

$$\max\left(0, c \cdot (1 - e^{\varepsilon - y_i})\right) \leq \max\left(0, d \cdot (1 - e^{\varepsilon - y_j})\right) \leq b \tag{2}$$

By assumption $\omega_1 \leq \omega_2$, the partial order Definition 10, and ascendingly-sorted $\mathcal{Y} = (y_i)_{i=1}^{m}$ s.t. $y_i < y_{i+1}$ of Definition 3, we know that for all y_i

$$\textstyle\sum_{y_j \in \{y_i, \dots, y_m, \infty\}} \omega_1(y_j) \leq \sum_{y_j \in \{y_i, \dots, y_m, \infty\}} \omega_2(y_j).$$

We now show the item of this lemma directly. By Definition 3, we know that

$$\begin{aligned}
&\delta(\varepsilon, \omega_2) \\
&= \sum_{\{y_i\}_{i=1}^{m}} \max\left(0, (1 - e^{\varepsilon - y_i}) \cdot \omega_2(y_i)\right) + \omega_2(\infty) \\
&= \sum_{\{y_i\}_{i=1}^{m-1}} \max\left(0, (1 - e^{\varepsilon - y_i}) \cdot \omega_2(y_i)\right) + \omega_2(\infty) \\
&\quad + \max\left(0, (1 - e^{\varepsilon - y_m}) \cdot \omega_2(y_m)\right) + \omega_1(\infty) - \omega_1(\infty)
\end{aligned}$$

by assumption $\omega_2(\infty) - \omega_1(\infty) \geq 0$ and by Equation (2) with $d = \omega_2(\infty) - \omega_1(\infty)$ we know that

$$\begin{aligned}
&\geq \sum_{\{y_i\}_{i=1}^{m-1}} \max\left(0, (1 - e^{\varepsilon - y_i}) \cdot \omega_2(y_i)\right) + \omega_1(\infty) \\
&\quad + \max\left(0, (1 - e^{\varepsilon - y_m}) \cdot (\omega_2(y_m) + \omega_2(\infty) - \omega_1(\infty))\right) \\
&= \sum_{\{y_i\}_{i=1}^{m-1}} \max\left(0, (1 - e^{\varepsilon - y_i}) \cdot \omega_2(y_i)\right) + \omega_1(\infty) \\
&\quad + \max\left(0, (1 - e^{\varepsilon - y_m}) \right. \\
&\qquad \left. \cdot (\omega_1(y_m) + \sum_{y_j \in \{y_m, \infty\}} \omega_2(y_j) - \omega_1(y_j))\right)
\end{aligned}$$

we apply Claim 1 with $k = m$

$$\begin{aligned}
&\geq \sum_{\{y_i\}_{i=1}^{m-2}} \max\left(0, (1 - e^{\varepsilon - y_i}) \cdot \omega_2(y_i)\right) + \omega_1(\infty) \\
&\quad + \max\left(0, (1 - e^{\varepsilon - y_{m-1}}) \right. \\
&\qquad \left. \cdot (\omega_1(y_{m-1}) + \sum_{y_j \in \{y_{m-1}, y_m, \infty\}} \omega_2(y_j) - \omega_1(y_j))\right) \\
&\quad + \max\left(0, (1 - e^{\varepsilon - y_m}) \cdot \omega_1(y_m)\right)
\end{aligned}$$

we repeat Claim 1 for k decreasing from $m - 1$ to 2

$$\begin{aligned}
&\geq \omega_1(\infty) + \max\left(0, (1 - e^{\varepsilon - y_1}) \right. \\
&\qquad \left. \cdot (\omega_1(y_1) + \sum_{y_j \in \{y_1, \dots, y_m, \infty\}} \omega_2(y_i) - \omega_1(y_i))\right) \\
&\quad + \sum_{\{y_i\}_{i=2}^{m}} \max\left(0, (1 - e^{\varepsilon - y_i}) \cdot \omega_1(y_i)\right)
\end{aligned}$$

since $\sum_y \omega_1(y) = \sum_y \omega_2(y) = 1$

$$\begin{aligned}
&= \sum_{\{y_i\}_{i=1}^{m}} \max\left(0, (1 - e^{\varepsilon - y_i}) \cdot \omega_1(y_i)\right) + \omega_1(\infty) \\
&= \delta(\varepsilon, \omega_1).
\end{aligned}$$

Claim 1. For all $y_k \in \{y_2, \dots, y_m\}$,

$$\begin{aligned}
&\max\left(0, (1 - e^{\varepsilon - y_{k-1}}) \cdot \omega_2(y_{k-1})\right) + \max\left(0, (1 - e^{\varepsilon - y_k}) \right. \\
&\qquad \left. \cdot (\omega_1(y_k) + \sum_{y_j \in \{y_k, \dots, y_m, \infty\}} \omega_2(y_j) - \omega_1(y_j))\right)
\end{aligned}$$

by assumption $\forall y_j \colon \omega_2(y_j) - \omega_1(y_j) \geq 0$ and by Equation (2) with $k-1 \leq k$ and $c = d = \sum_{y_j \in \{y_k,\dots,y_m,\infty\}} \omega_2(y_j) - \omega_1(y_j)$

$$\begin{aligned}
&\geq \max\big(0, (1-e^{\varepsilon - y_{k-1}}) \\
&\quad \cdot \big(\omega_2(y_{k-1}) + \sum_{y_j \in \{y_k,\dots,y_m,\infty\}} \omega_2(y_j) - \omega_1(y_j)\big)\big) \\
&\quad + \max\big(0, (1-e^{\varepsilon - y_k}) \cdot \omega_1(y_k)\big) \\
&= \max\big(0, (1-e^{\varepsilon - y_{k-1}}) \\
&\quad \cdot \big(\omega_1(y_{k-1}) + \sum_{y_j \in \{y_{k-1},\dots,y_m,\infty\}} \omega_2(y_j) - \omega_1(y_j)\big)\big) \\
&\quad + \max\big(0, (1-e^{\varepsilon - y_k}) \cdot \omega_1(y_k)\big).
\end{aligned}$$

□

1.5. Proof for Case 2 from Theorem 2. We show $\forall y_i \in \mathcal{Y} \colon \omega_{M,x_{a:b}} \underset{q\,\cdot\,\alpha/2}{\leq} \overline{\omega}_{M,x_{a:b},O^n}$. W.l.o.g. we assume that $y_j \leq y_i < y_{j+1}$ for all $y_j \in \mathcal{Y}^q \setminus \{y_q\}$. By Definition 10 and the definition of $\overline{\omega}$, we have

$$\overline{\omega}_{M,x_{a:b},O^n}(y_i) = \sum_{y' \geq y_i} \overline{\omega}_{M,x_{a:b},O^n}(y') = \overline{\varpi}_{M,x_{a:b},O^n}(y_i)$$

and by the step-function continuation of $\overline{\varpi}$ in Definition 9

$$= \overline{\varpi}_{M,x_{a:b},O^n}(y_j).$$

For the other side of the inequality, we have

$$\begin{aligned}
\omega_{M,x_{a:b}}(y_i) &= \sum_{y' \geq y_i} \omega_{M,x_{a:b}}(y') \\
&= \sum_{y' \geq y_j} \omega_{M,x_{a:b}}(y') - \sum_{y_j \geq y' > y_i} \omega_{M,x_{a:b}}(y') \\
&\leq \sum_{y' \geq y_j} \omega_{M,x_{a:b}}(y').
\end{aligned}$$

Thus, putting both sides of the inequality together, we have

$$\sum_{y' \geq y_j} \omega_{M,x_{a:b}}(y') \underset{q\,\cdot\,\alpha/2}{\leq} \overline{\varpi}_{M,x_{a:b},O^n}(y_j)$$

which holds with coverage probability $1-\alpha/2$ by Definition 9 for a given y_j. y_q is not contained in any interval, yet by the same argumentation where we have $y_i = y_q$, we conclude $\sum_{y' \geq y_q} \omega_{M,x_{a:b}}(y') \underset{q\,\cdot\,\alpha/2}{\leq} \overline{\varpi}_{M,x_{a:b},O^n}(y_q)$. For all $y_j \in \mathcal{Y}^q$, this inequality also holds by the Bonferroni correction since we use a coverage probability of $1 - q \cdot \alpha/2$.

We simplify the calculation of $\overline{\delta}(\varepsilon, \overline{\omega}_{M,x_{a:b},O^n})$ using the step-function continuation, i.e. $\overline{\varpi}_{M,x_{a:b},O^n}(y_i) = \overline{\varpi}_{M,x_{a:b},O^n}(y_j)$ in the interval $y_j \leq y_i < y_{j+1}$. Thus, we only have to sum over $\mathcal{Y}^q$ instead of $\mathcal{Y}$:

$$\begin{aligned}
&\overline{\delta}(\varepsilon, \overline{\varpi}_{M,x_{a:b},O^n}) \\
&= \overline{\varpi}_{M,x_{a:b},O^n}(\infty) + \sum_{(y_i)_{i=1}^{m-1}} \max\big(0, (1-e^{\varepsilon - y_{i+1}}) \\
&\qquad \cdot \big(\overline{\varpi}_{M,x_{a:b},O^n}(y_i) - \overline{\varpi}_{M,x_{a:b},O^n}(y_{i+1})\big)\big) \\
&= \overline{\varpi}_{M,x_{a:b},O^n}(\infty) + \sum_{(y_j)_{j=1}^{q-1}} \max\big(0, (1-e^{\varepsilon - y_{j+1}}) \\
&\qquad \cdot \big(\overline{\varpi}_{M,x_{a:b},O^n}(y_j) - \overline{\varpi}_{M,x_{a:b},O^n}(y_{j+1})\big)\big)
\end{aligned}$$

Since we upper bound the ε-attack advantage we upper bound the scale the privacy loss of the interval $(y_i, y_{i+1})_i$ with the rightmost element in the interval, i.e. $(1-e^{\varepsilon - y_{i+1}})$, for the lower bound we lower bound the scale with the leftmost element, i.e. $(1-e^{\varepsilon - y_i})$.

2. Example calculation of randomness tapes

In Figure 1 we have seen an example where we claimed that if u_0 sends their message to R_0, there are exactly 5 instances of randomness tapes that can make that happen and if u_0 sends their message to R_1, there are exactly 3 instances. Here we go through those instances.

Note that the randomness tapes contain the following pieces of information:

- Which message travels through which node in which round.
- Which of the senders u_1, u_2, u_3 has been randomly selected to send the message to R_{1-b}.

Each choice is made uniformly at random, i.e., each instance of randomness tapes occurs with exactly the same probability in our game. Not all of those lead to our observation from Figure 1.

For ease of readability, we describe the randomness tapes of each user as follows: [a,b,c,R], where a is their choice for the first node, b is their choice for the second node, c is their choice for the third node, and R is the recipient they send their message to.

If u_0 sends their message to R_0. These are all instances of randomness tapes that can make this happen:

1)	u_0	0	0	0	R_0
	u_1	0	0	0	$\mathsf{R}_{\mathsf{Noise}}$
	u_2	0	1	0	$\mathsf{R}_{\mathsf{Noise}}$
	u_3	1	1	1	R_1
2)	u_0	0	0	0	R_0
	u_1	0	0	0	$\mathsf{R}_{\mathsf{Noise}}$
	u_2	0	1	1	R_1
	u_3	1	1	0	$\mathsf{R}_{\mathsf{Noise}}$
3)	u_0	0	0	0	R_0
	u_1	0	1	0	$\mathsf{R}_{\mathsf{Noise}}$
	u_2	0	0	0	$\mathsf{R}_{\mathsf{Noise}}$
	u_3	1	1	1	R_1

4)	u_0	0	0	0	R_0
	u_1	0	1	1	R_1
	u_2	0	0	0	$\mathsf{R}_{\text{Noise}}$
	u_3	1	1	0	$\mathsf{R}_{\text{Noise}}$
5)	u_0	0	1	0	R_0
	u_1	0	0	0	$\mathsf{R}_{\text{Noise}}$
	u_2	0	0	0	$\mathsf{R}_{\text{Noise}}$
	u_3	1	1	1	R_1

As we can see, there are indeed exactly 5 instances of randomness tapes that lead to our observation.

If u_0 sends their message to R_1. We now list the instances of randomness tapes when u_0 sends their message to R_1. As we can see, they only differ in which of the messages was randomly selected to be the message sent to R_0.

1)	u_0	0	1	1	R_1
	u_1	0	0	0	R_0
	u_2	0	0	0	$\mathsf{R}_{\text{Noise}}$
	u_3	1	1	0	$\mathsf{R}_{\text{Noise}}$
2)	u_0	0	1	1	R_1
	u_1	0	0	0	$\mathsf{R}_{\text{Noise}}$
	u_2	0	0	0	R_0
	u_3	1	1	0	$\mathsf{R}_{\text{Noise}}$
3)	u_0	0	1	1	R_1
	u_1	0	0	0	$\mathsf{R}_{\text{Noise}}$
	u_2	0	0	0	$\mathsf{R}_{\text{Noise}}$
	u_3	1	1	0	R_0