# Differential Privacy in the Shuffle Model

Albert Cheu

**Abstract** Differentially private algorithms uncover information about a population while granting a form of individual privacy to any single member of the population. Research in differential privacy has primarily focused on one of two models. In the central model, a trusted aggregator runs a private algorithm. In the local model, owners of data run private algorithms themselves and an untrusted aggregator computes on the resulting messages. These models have inherent limitations. Solving statistical problems under local privacy demands many more samples than central privacy. On the other hand, central privacy is only possible if data owners grant an aggregator direct access to their data.

In this thesis, I introduce and study shuffle privacy, an intermediate model that strives for the benefits of both local and central privacy. Protocols in this model rely on a service that permutes messages uniformly at random, which makes communication anonymous. The model abstracts the PROCHLO analytics system developed at Google [Bittau et al., SOSP '17]. I describe shuffle protocols for statistical tasks like binary sums, histograms, and counting distinct elements. The protocols have provably better accuracy than local protocols and are also robustly private, since they ensure privacy in the face of drop outs. To complement these positive results, I also prove limitations of the model. Specifically, I show that robustly private shuffle protocols cannot learn parity or solve feature selection as accurately as centrally private algorithms.

# Dedication

I would like to thank Jonathan Ullman for being an excellent advisor and mentor. I owe him essentially the entirety of my knowledge of differential privacy. Indeed, I would not have been a part of the early days of shuffle privacy without him. I would also like to thank Adam Smith, Kobbi Nissim, and Ameya Velingker for recurring conversations and research discussions that elevated my work. Committee members Daniel Wichs and Abhi Shelat gave constructive feedback during my proposal and defense. And my work would not be possible without the great collaborators Maxim Zhilyaev, David Zeber, Victor Balcer, Matthew Joseph, and Jieming Mao.

I am grateful to have been a part of the 2019 Data Privacy program at the Simons Institute. Likewise, I would like to thank Aaron Roth for an extended visit at the University of Pennsylvania.

I have enjoyed my time as part of the Khoury College community. It was always nice to work in the company of fellow PhD. students. I am especially grateful to have Matthew Jagielski and Vikrant Singhal as roommates for five years.

Finally, I would like to thank my family for being a supportive presence in my life.

# Contents

# Chapter 1

# Introduction

Threats to an individual's privacy come in many forms, ranging from unwarranted government surveillance to corporate tracking of consumption patterns. This thesis focuses on the privacy in the context of aggregate statistics. Consider, for example, the developers of a keyboard app who wish to identify common words and their mispellings. Each user of the app contributes a row of data to a large table and the app developers want to obtain word-usage statistics from the table in some rigorously private way.

To do so, the app developers can design *differentially private* algorithms. Defined by Dwork, McSherry, Nissim, and Smith [28], such an algorithm ensures that the odds of any output do not change significantly when a row is removed or swapped from the data table. In terms of our example, the privately collected statistics could help the keyboard app autocorrect "teh" to "the" but the odds of that occurring are not that much different if Alice had changed the contents of her text with Bob.

In order to create their private algorithms, the developers of the keyboard app can and must add noise to the statistics. For instance, if they want to report the frequency of "teh" while satisfying differential privacy, they can add binomial noise to the raw count. The magnitude of this noise cannot be too small. Work by Dinur and Nissim [25] shows that an adversary can reconstruct individuals' information from too many accurate answers to database queries. In a nutshell, noise is a necessary condition for privacy.[1] The challenge is to inject enough noise to deter attacks but not so much to significantly impact accuracy.

Implicit in the above discussion is the assumption that the app developer is *trusted*, meaning that data contributors believe it only runs differentially private algorithms. This setting has naturally been called the *trusted curator model* or the *central model*. But even when the app developer promises to run a differentially private algorithm, people may still hesitate to contribute their data. For example, they may not believe the interests of the aggregator are in line with their privacy, or they may simply worry that the aggregator is vulnerable to security breaches.

*Local privacy* [60, 34, 28, 49] is an alternative to central privacy that aims to overcome these concerns. Here, each individual executes a differentially private algorithm on their datum, rather than relying on an outside entity. Then they send a message containing the algorithm's output to a central server. In our running example, each user's smartphone can maintain a bit that indicates whether they used "teh" and, to ensure differential privacy, the developer receives randomized versions of the bits. Each user could, for example, flip their bit with probability 0.48. But because each user introduces a nontrivial amount of noise, the developer's estimate is much worse than if it had direct access to the bits. Specifically, the variance of the estimate must be linear in the number of users for any locally private algorithm [14, 21],

[1] This theoretical result is backed by work due to Garfinkel, Abowd, and Martindale [36], who instantiate successful reconstruction attacks against publications of prior Census statistics.

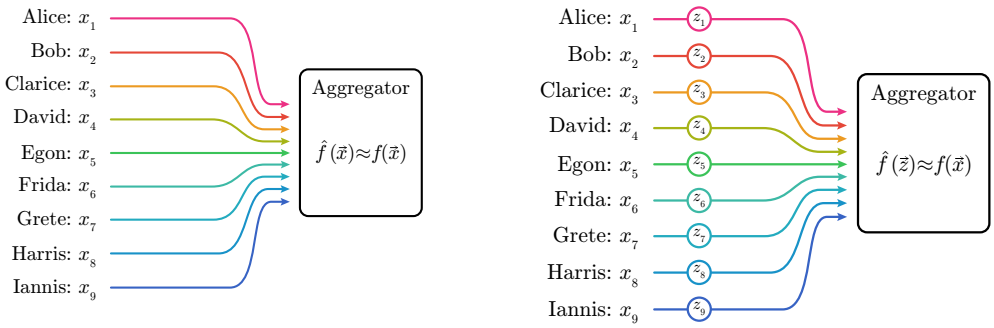


Figure 1.1: Schematic of central privacy (left) and local privacy (right). In central privacy, the aggregator runs a differentially private algorithm on user data. In local privacy, each user runs a private algorithm on their own data.

but there is a centrally private algorithm without such a dependence [28]. There is a long line of work that shows similar limitations hold for a host of estimation and learning problems [12, 1, 46].

These negative results motivate the exploration of intermediate trust models that can achieve some of the "best of both worlds". Specifically,

> Can we achieve the accuracy that is possible with centrally private algorithms from a trust assumption that is close to locally private protocols?

In this thesis, we introduce and formally study the *shuffle model* of differential privacy (henceforth the shuffle model). Like the local model, users produce messages by way of a local randomizer on their data but now they trust some entity to apply a uniformly random permutation on all user messages. We assume that the adversary's view is limited to that permutation, so no message can be linked back to its sender. We also restrict attention to one round of non-interactive communication, meaning that each user generates one batch of messages independently of other users. A pictorial representation of the model is in Figure 1.2.

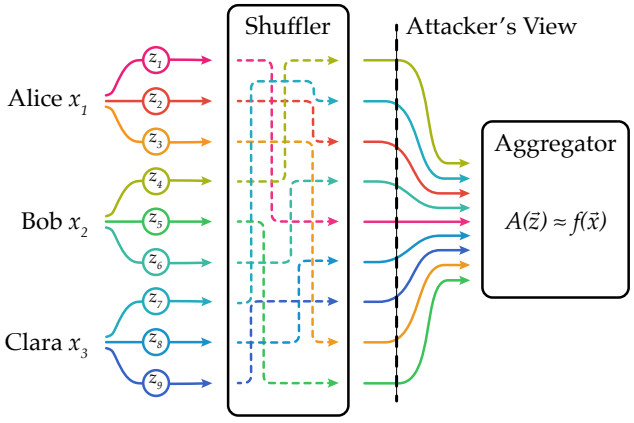


Figure 1.2: Schematic of shuffle privacy.

Aside from being the focus of this thesis, the shuffle model has inspired a substantial amount of parallel work in recent years. We give an overview in Section 1.2.

**Reasons to study the shuffle model.** Given the infinitude of possible models, we take a moment to justify the focus on the shuffle model. Firstly, the shuffle model is an abstraction of the PROCHLO system developed by engineers and researchers at Google [15], so understanding the power and limits of the shuffle model will give insight into what is possible with systems like PROCHLO.

Secondly, the shuffle model can be interpreted as a restricted form of *multi-party computation* [61, 62, 41] that admits efficient implementations. An MPC protocol is a cryptographic simulation of a central model algorithm on data distributed across parties. The protocol guarantees that the only information attainable by any party is the output of the simulated algorithm. Beginning with work by Dwork, Kenthapadi, McSherry, Mironov, and Naor [27], a line of work explores the possibility of simulating centrally private algorithms using MPC without any trusted party [54, 42, 32, 53, 16]. In principle, this class of protocol would have precisely the same accuracy as central privacy. There are some hurdles in this approach, however: current methods of simulating arbitrary algorithms have large computation and communication costs, making them difficult to scale and maintain. In contrast, we can implement a shuffler with textbook onion routing. So shuffle protocols capture a rich subclass of MPC protocols that can be implemented relatively efficiently.

Lastly, the shuffle model compares favorably to another, more studied subclass of MPC called the secure aggregation model [5, 55, 42, 56, 59, 47]. There, protocols assume the existence of a primitive that securely performs vector addition over a finite field instead of uniformly random permutation. As we explain in Section 1.3.2, the literature has shown it is possible to instantiate this primitive in the shuffle model. This transformation implies that private protocols in the shuffle model are at least as a strong as those in the secure aggregation model.

**Understanding the power of the shuffle model.** In the shuffle model, observe that users have two layers of protection: the anonymity offered by the shuffle and the randomness in the messages. The key result of this assumption is that each user in a shuffle protocol can introduce less noise than in a local protocol with the same privacy guarantee and only a mildly stronger trust assumption.

To make this consequence concrete, we return to the scenario where the app developer wishes to privately estimate the frequency of "teh." Suppose each user's phone sends two messages to the shuffler: their indicator bit followed by a bit set to 1 with probability $p$ and 0 with probability $1 - p$. Because the shuffler outputs a random permutation of the message bits, an adversary gleans exactly the same information about a user from the messages as contained in the sum of the messages. It therefore suffices to ensure that the sum of the messages is differentially private. When there are $n$ users, this sum has binomial noise with variance $np(1 - p)$. And when the app sets $p \propto 1/n$, we can achieve error with no explicit dependence on $n$ just like in the central model.

In addition to improved privacy-accuracy tradeoffs, most shuffle protocols also exhibit *robust differential privacy*. This means a user's privacy is only slightly degraded when a small fraction of peers drop out of the protocol. Note that locally private protocols guarantee an extreme form of robustness, since a user's privacy is completely independent of the behavior of their peers.

## 1.1 Overview of Contributions and Techniques

Following this introduction, this thesis consists of three major chapters. Our results are stated for $(\varepsilon, \delta)$-differential privacy, where $\varepsilon$ and $\delta$ are parameters between 0 and 1 that bound the advantage given to an adversary (see Definition 1.3.1).

### 1.1.1 Chapter 2: Novel Shuffle Protocols.

This chapter describes shuffle protocols for a variety of statistical tasks. All satisfy robust differential privacy and have better accuracy than competing local protocols. We summarize the techniques here, then present a table of the asymptotic results.

Section 2.1 describes a binary sum protocol $\mathcal{P}_{\mathsf{SYM}}$. Like the sketch in the preceding section, this protocol mixes noise bits with data bits to ensure privacy via binomial noise. The difference is that $\mathcal{P}_{\mathsf{SYM}}$'s estimate is symmetrically distributed about the true sum, hence the name. The analysis of the protocol is followed by applications to mean estimation and feature selection.

When each user has one of $d > 2$ different values, Section 2.2 describes a shuffle protocol that computes $d$-bin histograms. To obtain the frequency estimates, the protocol executes a binary sum protocol $\mathcal{P}_{\mathsf{ZSUM}}$ multiple times in parallel. $\mathcal{P}_{\mathsf{ZSUM}}$ deterministically outputs 0 when the input is all 0 and otherwise adds noise with low bias and variance (Section 2.2.2). In the context of histograms, this property allows us to perform a union bound over a number of executions that is independent of the domain size (i.e. independent of size of dictionary). In Section 2.3, we show that the histogram result implies that we can identify the support of a distribution under robust shuffle privacy with significantly fewer samples than non-interactive local privacy.

In Section 2.4, we show how to count the number of distinct elements in the shuffle model. To do so, we first show how to privately compute the bit that indicates whether or not an element $j$ is present in a dataset. Like our histogram protocol, our distinct elements protocol executes this primitive $d$ times in parallel, once for each possible element $j$. The protocol reports the sum of all $d$ bits.

In Section 2.5, we show how to test if a data distribution is uniform or far from uniform. This is done by post-processing a private histogram. To ease a step in the analysis, we replace $\mathcal{P}_{\mathsf{ZSUM}}$ with our original binary sum protocol $\mathcal{P}_{\mathsf{SYM}}$.

We summarize our positive results in Table 1.1 and contrast them with lower bounds in the local model. Note that lower bounds in the local model are typically derived for the case where $\delta = 0$, but prior work implies that such lower bounds carry over to $\delta = O(1/n)$ (see Lemma 1.3.10).

### 1.1.2 Chapter 3: The Limits of Robust Shuffle Privacy.

To complement our protocols, we derive impossibility results. Specifically, it is not possible to perform uniformity testing (Section 3.2) or count distinct elements (Section 3.1) under robust shuffle privacy with as little noise as under central privacy. This is also the case for feature selection (Section 3.3), learning parity functions (Section 3.4), and other tasks. We summarize these impossibility results and compare with centrally private algorithms in Table 1.2.

These lower bounds rely on a key structural result: every robustly private shuffle protocol implies an online algorithm with the same privacy parameters and approximately the same accuracy. The online algorithm simulates the shuffle protocol by running the local randomizer on each incoming data value. At any point in time, the algorithm's internal state is a shuffled set of messages generated by the randomizer up to that point. Because the shuffle protocol is robustly private, this intuitively grants some level of differential privacy to any single state. This intuition breaks down for user 1, since the first state only contains messages from that user. To patch this, the algorithm initializes its state by running the randomizer many times on dummy data points. This has the effect of shifiting the underlying distribution but to a bounded degree.

Given this transformation from the shuffle model to the online model, it suffices to invoke lower bounds for differentially private online algorithms. Prior work contains such results for uniformity testing and counting distinct elements. For the other problems we consider, we derive new ones using a nontrivial extension of techniques in the local privacy literature.

Table 1.1: Comparison of upper bounds under robust shuffle privacy with lower bounds under local privacy. To simplify presentation, we assume success probability 99/100. $d$ and $\alpha$ are dimension and error parameters, respectively. $\ell, h$ are specific to pointer chasing and multi-party pointer jumping.

| | | $(\varepsilon,\delta)$-Local Privacy | $(\varepsilon,\delta)$-Robust Shuffle Privacy (This thesis) |
|---|---|---|---|
| Additive Error of | Binary Sums | $\Omega\left(\frac{1}{\varepsilon}\sqrt{n}\right)$ [14, 21] | $O\left(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}}\right)$ Thm 2.1.4 |
| | Distinct Elements | $\Omega(n)$ [22] ($d = n$) | $O\left(\frac{1}{\varepsilon}\cdot\min(\sqrt{d}, n^{2/3})\right)$ Thm 2.4.2 |
| $\ell_\infty$ Error of | Histograms | $\Omega\left(\frac{1}{\varepsilon}\sqrt{n\log d}\right)$ [12] | $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$ Thm 2.2.2 ($\delta = O(1/n)$) |
| Sample Complexity of | Mean Estimation (distributions over $[0,1]$) | $\Omega\left(\frac{1}{\alpha^2\varepsilon^2}\right)$ [14, 21] | $O\left(\frac{1}{\alpha^2}+\frac{1}{\alpha\varepsilon}\sqrt{\log\frac{1}{\delta}}\right)$ Thm 2.1.9 |
| | Feature Selection | $\Omega(\frac{d\log d}{\alpha^2\varepsilon^2})$ [58] | $\tilde{O}\left(\frac{\log d}{\alpha^2}+\frac{\sqrt{d}}{\alpha\varepsilon}\log\frac{1}{\delta}\right)$ Thm 2.1.14 |
| | Uniformity Testing | $\Omega\left(\frac{d}{\alpha^2\varepsilon^2}\right)$ [1] | $O\left(\left[\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}+\frac{\sqrt{d}}{\alpha^2}+\frac{\sqrt{d}}{\alpha\varepsilon}\right]\sqrt{\log\left(\frac{1}{\delta}\right)}\right)$ Thm 2.5.2 |
| | Multi-Party Pointer Jumping | $\Omega\left(\frac{h^3}{\varepsilon^2\log h}\right)$ [45] | $O\left(h\log h\cdot\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$ Thm 2.3.4 ($\delta < 1/200h$) |
| | Pointer Chasing | $\Omega(\ell)$ [46] | $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$ Thm 2.3.6 |

Table 1.2: Comparison of our impossibility results for robust shuffle privacy with centrally private algorithms. As before, $d$ and $\alpha$ are dimension and error parameters, respectively. $*$ indicates that $\delta \log \frac{d}{\delta} \ll \alpha^2\varepsilon^2/d$ and $**$ indicates that $\delta \log \binom{d}{\leq k}/\delta \ll \alpha^2\varepsilon^2/\binom{d}{\leq k}$.

| | | $(\varepsilon,\delta)$-Robust Shuffle Privacy (This thesis) | $\varepsilon$-Central Privacy |
|---|---|---|---|
| Additive Error of | Distinct Elements | $\Omega\left(\sqrt{\frac{d}{\varepsilon}}+\frac{1}{\varepsilon}\right)$ Thm 3.1.1 ($n \geq 2d$) | $O\left(\frac{1}{\varepsilon}\right)$ [28] (Laplace mech.) |
| Sample Complexity of | Uniformity Testing | $\Omega\left(\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}+\frac{\sqrt{d}}{\alpha^2}+\frac{1}{\alpha\varepsilon}\right)$ Thm 3.2.1 ($\delta=0$) | $O\left(\frac{\sqrt{d}}{\alpha^2}+\frac{\sqrt{d}}{\alpha\varepsilon}+\frac{d^{1/3}}{\alpha^{4/3}\varepsilon^{2/3}}+\frac{1}{\alpha\varepsilon}\right)$ [4] |
| | Feature Selection | $\Omega\left(\frac{\sqrt{d}}{\alpha\varepsilon}\right)$ Thm 3.3.1 $*$ | $O\left(\frac{\log d}{\alpha^2}+\frac{\log d}{\alpha\varepsilon}\right)$ [50] (Exp. mech.) |
| | Simple Hypothesis Testing | $\Omega\left(\frac{\sqrt{d}}{\alpha\varepsilon}\right)$ Thm 3.3.15 $*$ | $O(\log d)$ [17] |
| | 1-Sparse Mean Est. | $\Omega\left(\frac{\sqrt{d}}{\alpha\varepsilon}\right)$ Thm 3.3.18 $*$ | $O(\log d)$ [Folklore] |
| | Parity Release | $\Omega\left(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon\right)$ Thm 3.3.21 $**$ | $\tilde{O}(\sqrt{d}\log\binom{d}{\leq k})$ [43] |
| | Parity Learning | $\Omega\left(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon\right)$ Thm 3.4.3 $**$ | $O(\log\binom{d}{\leq k})$ [49] |

### 1.1.3 Chapter 4: Single-Message Shuffle Privacy

In the third and final chapter, we shift focus from robustly private protocols to single-message protocols. As the name suggests, these are shuffle protocols where each user sends exactly one message to the shuffler. The canonical protocol we study is binary randomized response $\mathcal{P}_{\mathsf{RR}}$ (Section 4.1). Unlike $\mathcal{P}_{\mathsf{SYM}}$, each user only sends one bit but it achieves the same error. We complement this protocol with a structural result: every single-message shuffle protocol implies a local protocol with exactly the same accuracy but loosened privacy guarantees (Section 4.2). This enables the derivation of lower bounds for single-message shuffle privacy. In particular, these lower bounds show that the model is noisier than the central model.

One way to make single-message shuffle protocols is to study how well shuffling amplifies the privacy guarantee of a generic locally private protocol. In Section 4.3, we show that existing analysis is essentially optimal.

## 1.2 Related Work

Many other researchers have explored the intersection of shuffling and differential privacy and we give a bird's-eye view here.

**Privacy Amplification by Shuffling.** Shuffling the messages produced by differentially private local randomizers can improve the privacy guarantee offered to the users. For a crude intuition, suppose the adversary possesses an optimal strategy to reconstruct data from the output of randomizer $\mathcal{R}$. When given a shuffled set of such messages, the adversary could execute this strategy on every message but will have to map its guesses to users. A line of work has quantified the relationship between the $\varepsilon$ privacy parameter of the shuffle protocol and the number of users $n$. These *amplification-by-shuffling lemmas* are derived by Erlingsson, Feldman, Mironov, Raghunathan, Talwar, and Thakurta [33], Balle, Bell, Gascón, and Nissim [11], culminating with the recent work of Feldman, McMillan, and Talwar [35]. As we show in Section 4.3, their amplification lemma turns out to be nearly optimal.

**Shuffle Protocols.** There has been much work on shuffle protocols for summation. In their first work on the shuffle model, Balle et al. [11] gave a single-message protocol for sums of values in $[0,1]$. In follow-up work, they show how to adapt work by Ishai et al. to simulate the discrete Laplace mechanism in the shuffle model [9]. They later show how to reduce the number of messages per user in this protocol [10]. Ghazi, Manurangsi, Pagh, and Velingker [39] arrive at the same number of messages as [10] (up to constants) using different proof techniques. Ghazi, Golowich, Kumar, Manurangsi, Pagh, and Velingker [37] give the first shuffle protocol that satisfies pure differential privacy while also having the same magnitude of error as the Laplace mechanism. This thesis focuses on sums of values in $\{0,1\}$ instead of more general $[0,1]$ sums because a protocol for the latter implies one for the former. we summarize the binary sum results in Table 2.1.

Ghazi, Golowich, Kumar, Pagh, and Velingker [38] give two histogram protocols with polylogarithmic error and communication complexity. We compare these results with ours in Table 2.2.

**Lower Bounds for Shuffle Protocols.** To match their upper bound, Balle et al. [11] give a lower bound on the error of any single-message protocol for computing a sum of values in $[0,1]$ .

Chen, Ghazi, Kumar, and Manurangsi give lower bounds when users can only send $m$ messages in a shuffle private protocol [22]. They show that the local randomizer must satisfy a property they call *pseudo-locally private* and then give lower bounds for such protocols. This is a generalization of the $m = 1$ case considered in Section 4.2. Concurrently, Beimel et al. [13] also give lower bounds for

any message-limited shuffle protocol using an information-theoretic argument. Unlike the technique employed throughout Chapter 3, these arguments cannot give lower bounds for protocols with a large number of messages.

**Variations of Shuffle Model.** The shuffle model is derived from the Encode-Shuffle-Analyze (ESA) architecture introduced by Bittau et al. [15]. In addition to applying a random permutation, their shuffler drops a message if it does not have many duplicates in the set of messages. In order to focus on the power of a relatively weak primitive, the work in this thesis is limited to the study of a shuffler that only applies a permutation. We leave a rigorous analysis of thresholding for future work.

The work by Erlingsson et al. [33] and Feldman et al. [35] consider a model where a server communicates with each user once in a uniformly random order. This model captures the execution of sequentially interactive local protocols on a shuffled set of users, where the randomizer of user $i$ adapts to the messages of users $1, \ldots, i-1$. The model considered by this thesis does not permit interactivity but does allow multiple messages from each user.

Other work considered protocols that use shufflers multiple times. For example, Beimel, Haitner, Nissim, and Stemmer [13] leverage this functionality to instantiate generic MPC, meaning that protocols that use the shuffler twice can implement arbitrary central-model algorithms. Earlier work by Ishai, Kushilevitz, Ostrovsky, and Sahai [44] also use multi-round shuffling for the purposes of MPC. They consider a variant of the shuffler where a party can reply to the sender of a message without knowing their identity. In our work, we only focus on protocols where users communicate once to the shuffler.

**Robust Distributed DP.** We remark that Ács and Castelluccia [5] put forth a notion of robust differential privacy: a distributed protocol should satisfy a target level of differential privacy whenever there are at least a certain fraction of honest users. But to satisfy our definition, a protocol satisfies a target level of differential privacy when all users are honest and the guarantee must simply degrade smoothly as the fraction of honest users shrinks.

## 1.3 Technical Background

We reserve boldface letters to denote probability distributions and vector notation $\vec{\cdot}$ for ordered sequences of objects. Throughout this work, we use the notation $[d] := \{1, 2, \ldots, d\}$.

$\mathcal{X}$ denotes a *data universe* and a *dataset* $\vec{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$ is an ordered tuple of $n$ rows from the universe. Each row belongs to one individual called a *user*. Two datasets $\vec{x}, \vec{x}' \in \mathcal{X}^n$ are considered *neighbors* if they differ in at most one row. This is denoted as $\vec{x} \sim \vec{x}'$.

Differential privacy is defined in the seminal work by Dwork, McSherry, Nissim, and Smith [28].

**Definition 1.3.1** (Differential Privacy [28])**.** An algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{Z}$ satisfies $(\varepsilon, \delta)$-*differential privacy* if, for every $\vec{x} \sim \vec{x'}$ and every $Z \subset \mathcal{Z}$,

$$\mathbb{P}\left[\mathcal{M}(\vec{x}) \in Z\right] \leq e^{\varepsilon} \cdot \mathbb{P}\left[\mathcal{M}(\vec{x}') \in Z\right] + \delta \tag{1.1}$$

We emphasize that the probability is over the algorithm and not in the inputs. When $\delta > 0$, we say $\mathcal{M}$ satisfies *approximate* differential privacy. When $\delta = 0$, $\mathcal{M}$ satisfies *pure* differential privacy and we omit the $\delta$ parameter.

Because the above definition assumes that the algorithm $\mathcal{M}$ has "central" access to compute on the entire raw dataset, we call this *central privacy* for brevity. We remark that we will typically prove theorems for $\varepsilon \leq 1$. In addition to being a reasonable level of privacy, it simplifies the presentation. For example, $e^{\varepsilon} = 1 + O(\varepsilon)$ and $\frac{\exp(\varepsilon)+1}{\exp(\varepsilon)-1} = O(1/\varepsilon)$ in this regime.

We note that it is possible to change (1.1) to some other notion of distance between distributions but this choice satisfies *group privacy* and *composition*. We define these notions below:

**Fact 1.3.2** (Group Privacy)**.** *For any $(\varepsilon, \delta)$-differentially private algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Z}$, if $\vec{x}$ and $\vec{x}'$ differ on $d$ positions, then for any $Z \subseteq \mathcal{Z}$ and $d \in \mathbb{N}$,*

$$\mathbb{P}\left[\mathcal{M}(\vec{x}) \in Z\right] \leq e^{d\varepsilon} \cdot \mathbb{P}\left[\mathcal{M}(\vec{x}') \in Z\right] + d\exp((d-1)\varepsilon)\delta.$$

**Fact 1.3.3** (Basic Composition)**.** *For any $(\varepsilon, \delta)$-differentially private algorithms $\mathcal{M}_1$ and $\mathcal{M}_2$, the algorithm $\mathcal{M}_3$ defined by $\mathcal{M}_3(\vec{x}) = (\mathcal{M}_1(\vec{x}), \mathcal{M}_2(\vec{x}))$ is $(2\varepsilon, 2\delta)$-differentially private.*

**Fact 1.3.4** (Advanced Composition)**.** *For any $(\varepsilon, \delta)$-differentially private algorithms $\mathcal{M}_1, \ldots, \mathcal{M}_d$, the algorithm $\overline{\mathcal{M}}$ defined by $\overline{\mathcal{M}}(\vec{x}) = (\mathcal{M}_1(\vec{x}), \ldots, \mathcal{M}_k(\vec{x}))$ is $\left(\varepsilon \cdot \left(\sqrt{2d\ln(1/\delta)} + (e^{\varepsilon} - 1) \cdot d\right), \delta \cdot (d+1)\right)$-differentially private.*

Another key property of differential privacy is closure under post-processing. This means that any computation based solely on the output of a differentially private function does not affect the privacy guarantee.

**Fact 1.3.5.** *For $(\varepsilon, \delta)$-differentially private algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{Z}$ and arbitrary function $f : \mathcal{Z} \to \mathcal{Z}'$, $f \circ \mathcal{M}$ is $(\varepsilon, \delta)$-differentially private.*

Proofs of these facts appear in the survey of Dwork and Roth [30].

A class of centrally private algorithms are additive noise mechanisms. These provide private estimates of 1-sensitive functions $f : \mathcal{X}^n \to \mathbb{R}$, where the inequality $|f(\vec{x}) - f(\vec{x}')| \leq 1$ holds for all $\vec{x} \sim \vec{x}' \in \mathcal{X}^n$. The definition can be generalized to $\Delta$-sensitivity for any $\Delta > 0$, though this work will only focus on the case where

The canonical example of an additive noise mechanism is the *Laplace mechanism* by Dwork et al.

**Lemma 1.3.6** (From [28])**.** *Fix any 1-sensitive function $f : \mathcal{X}^n \to \mathbb{R}$ and any $\varepsilon > 0$. Let $\mathcal{M}_{f,\varepsilon}$ denote the algorithm that samples $\eta \sim \mathbf{Lap}(1/\varepsilon)$ and outputs $f(\vec{x}) + \eta$. $\mathcal{M}_{f,\varepsilon}$ is $\varepsilon$-differentially private.*

Another example is the *binomial mechanism*. Although it has been studied previously (e.g. [27, 38]), we present the mechanism's privacy guarantees in a form that is more amenable to our arguments.

**Lemma 1.3.7.** *Let $f : \mathcal{X}^n \to \mathbb{Z}$ be a 1-sensitive function and fix any $\delta < 2e^{-9}$. For any $m \in \mathbb{N}$ and $p \in (0,1)$, let $\mathcal{M}_{f,m,p}$ denote the algorithm that samples $\eta \sim \mathbf{Bin}(m,p)$ and outputs $f(\vec{x}) + \eta$. If $m \cdot \min(p, 1-p) > 13\ln\frac{2}{\delta}$, $\mathcal{M}_{f,m,p}$ is $(\varepsilon(m,p), \delta)$-differentially private, where*

$$\varepsilon(m,p) := \ln\left(1 + \sqrt{\frac{13\ln\frac{2}{\delta}}{m\min(p,1-p)}}\right) < \sqrt{\frac{13\ln\frac{2}{\delta}}{m\min(p,1-p)}}.$$

We prove this lemma in Appendix A.1.

### 1.3.1 Local Protocols

A user may be skeptical of implementations of centrally private algorithms. They may fear improper execution or they may be wary of the fact that those executing the algorithm can inspect the data inputs. Potential breaches by external parties are yet another source of concern. In the extreme, no user trusts any other party with protecting their data; here, we model the dataset as a distributed object where each of $n$ users holds a single row. Each user $i$ provides their data point as input to a randomizing function $\mathcal{R}$ and publishes the outputs for some analyzer to compute on.

**Definition 1.3.8** (Local Model [60, 34])**.** A protocol $\mathcal{P}$ in the *local model* consists of two randomized algorithms:

- A randomizer $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ mapping data to a message.
- An analyzer $\mathcal{A} : \mathcal{Y}^n \to \mathcal{Z}$ that computes on a vector of messages.

We define its execution on input $\vec{x} \in \mathcal{X}^n$ as

$$\mathcal{P}(\vec{x}) := \mathcal{A}(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n)).$$

We assume that $\mathcal{R}$ and $\mathcal{A}$ have access to an arbitrary amount of public randomness.

To protect each user's data, local differential privacy imposes the privacy constraint on $\mathcal{R}$.

**Definition 1.3.9** (Local Differential Privacy [28, 49])**.** A local protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is $(\varepsilon, \delta)$-*differentially private* if $\mathcal{R}$ is $(\varepsilon, \delta)$-differentially private. The privacy guarantee is over the internal randomness of the users' randomizers and not the public randomness of the protocol.

For brevity, we typically call these protocols "locally private." We remark that the local privacy literature includes *interactive* protocols, where a message sent by a user can depend on prior messages (e.g. distributed stochastic gradient descent). But our lower bounds will only use results from non-interactive local privacy and our protocols will only be compared with non-interactive local protocols. For this reason, we will omit the term "non-interactive" and simply use "local protocol" and "local privacy."

Using results from Kairouz, Oh, and Viswanath [48] and Murtagh and Vadhan [52, Lemma 3.2], we can show that it is without much loss of generality to consider only the case where $\delta = 0$:

**Lemma 1.3.10.** *If local randomizer $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ is $(\varepsilon, \delta)$-differentially private, then there is a local randomizer $\mathcal{R}'$ that is $(2\varepsilon, 0)$-differentially private such that*

$$\forall x \in \mathcal{X} \quad \mathrm{d}_{\mathrm{TV}}(\mathcal{R}(x), \mathcal{R}'(x)) \leq \delta$$

Refer to Appendix A.1 for a proof.

### 1.3.2 Shuffle Protocols

We now arrive at the model of privacy that is the focus of this work. First rigorously defined in joint work with Smith Ullman Zeber and Zhilyaev [23], it is a formalization of work done by Bittau et al. [15].

We begin with a preliminary version of the shuffle model. This version, *the single-message shuffle model*, is a straightforward relaxation of the local model: each user executes $\mathcal{R}$ on their data to produce a message as before, but now they trust some entity to perform a secure shuffle on all $n$ user messages. An adversary's view is therefore limited to a uniformly random permutation of the messages, so no message can be linked back to its sender.

In this thesis, we allow *each user to send any number of messages* to the shuffler. The shuffling prevents messages from the same sender from being linked with one another.

**Definition 1.3.11** (Shuffle Model [15, 23])**.** A protocol $\mathcal{P}$ in the *shuffle model* consists of three randomized algorithms:

- A *randomizer* $\mathcal{R} : \mathcal{X} \to \mathcal{Y}^*$ mapping a datum to a (possibly variable-length) vector of messages.
- A *shuffler* $\mathcal{S} : \mathcal{Y}^* \to \mathcal{Y}^*$ that applies a uniformly random permutation to the messages in its input.
- An *analyzer* $\mathcal{A} : \mathcal{Y}^* \to \mathcal{Z}$ that computes on a permutation of messages.

As $\mathcal{S}$ is the same in every protocol, we identify each shuffle protocol by $\mathcal{P} = (\mathcal{R}, \mathcal{A})$. We define its execution by $n$ users on input $\vec{x} \in \mathcal{X}^n$ as

$$\mathcal{P}(\vec{x}) := \mathcal{A}(\mathcal{S}(R(x_1), \ldots, R(x_n))).$$

As with local privacy, we grant the parties an arbitrary amount of public randomness. Importantly, we also allow $\mathcal{R}$ and $\mathcal{A}$ to have parameters that depend on $n$.

The following is a definition of differential privacy in this model.

**Definition 1.3.12** (Differential Privacy for Shuffle Protocols[23])**.** A shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is $(\varepsilon, \delta)$-*differentially private for $n$ users* if the algorithm $(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}) := \mathcal{S}(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$ is $(\varepsilon, \delta)$-differentially private. The privacy guarantee is over the internal randomness of the users' randomizers and not the public randomness of the shuffle protocol.

For brevity, we typically call these protocols "shuffle private."

Note that Definition 1.3.12 assumes all $n$ users follow the protocol. In the less-optimistic scenario investigated by Balcer et al. [8], only a $\gamma$ fraction of the users are *honest* and the rest are *corrupted*. The honest users perform the randomization $\mathcal{R}$ as intended but the corrupted users attempt to degrade the privacy guarantees for the honest users. In the worst case that we consider, the corrupted users can collude and transmit arbitrary messages to $\mathcal{S}$. Ideally, honest users should be guaranteed some level of differential privacy regardless of the attack.

One attack is to simply drop out: $\mathcal{S} \circ \mathcal{R}^{\gamma n}$ denotes the execution of the protocol in this case (assuming that $\gamma n \in \mathbb{N}$). Because $\mathcal{R}$ takes $n$ as a parameter but does not have access to $\gamma$, the privacy parameters actually offered by the protocol may depend on $\gamma$ due to miscalibration. In an ideal protocol, the privacy parameters are bounded by functions $\tilde{\varepsilon}(\gamma), \tilde{\delta}(\gamma)$ which smoothly decrease as $\gamma$ approaches 1.

Dropping out is the worst that corrputed users can do with respect to differential privacy. This comes from differential privacy's resilience to post-processing (Fact 1.3.5): if $\mathcal{S} \circ \mathcal{R}^{\gamma n}$ is already $(\varepsilon, \delta)$-differentially private, then the algorithm that shuffles the messages produced by $\mathcal{S} \circ \mathcal{R}^{\gamma n}$ with the corrupted users' messages is also $(\varepsilon, \delta)$-differentially private. Hence, the following *robust* variant of shuffle privacy focuses on drop-out attacks without loss of generality.

**Definition 1.3.13** (Robust Differential Privacy [8])**.** Fix continuous and non-increasing functions $\tilde{\varepsilon}, \tilde{\delta}$ such that $0 < \tilde{\varepsilon}(\gamma) < \infty$ and $0 < \tilde{\delta}(\gamma) < 1$ for all $\gamma \in [1/2, 1]$.[2] A shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is $(\tilde{\varepsilon}, \tilde{\delta})$-*robustly differentially private for $n$ users* if, for all $\gamma \in [1/2, 1]$ such that $\gamma n \in \mathbb{N}$, the algorithm $\mathcal{S} \circ \mathcal{R}^{\gamma n}$ is $(\tilde{\varepsilon}(\gamma), \tilde{\delta}(\gamma))$-differentially private.

As with the Definition 1.3.12, we often shorthand Definition 1.3.13 as "robust privacy." We remark that it is natural to design protocols where a target level of privacy holds when all users are honest ($\gamma = 1$) and privacy degrades only slightly when there is a honest majority ($\gamma = 1/2$). We use the following shorthand to cover this case:

**Definition 1.3.14.** A shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is $(\varepsilon, \delta)$-robustly private for $n$ users if there are continuous and non-increasing functions $\tilde{\varepsilon}, \tilde{\delta}$ such that

$$\tilde{\varepsilon}(1) = \varepsilon, \ \tilde{\delta}(1) = \delta,$$
$$\tilde{\varepsilon}(1/2) = O(\varepsilon), \ \tilde{\delta}(1/2) = O(\delta),$$

and $\mathcal{P}$ satisfies $(\tilde{\varepsilon}, \tilde{\delta})$-robust privacy for $n$ users.

Note that we have defined robustness with regard to privacy rather than accuracy. A robustly private shuffle protocol promises its users that their privacy will not suffer much from a limited fraction of malicious users. But it does not make any guarantees about the accuracy of the protocol; accuracy statements are made under the assumption that all users follow the protocol ($\gamma = 1$).

### Comparison with Secure Aggregation

Suppose we replace the shuffler with a trusted primitive that computes sums, which we denote by $\hat{\mathcal{S}}_d$. Each user in this *secure aggregation model* sends one message that is a numerical vector of dimension $d$

[2] We could change 1/2 to a parameter $\tau \in (0, 1)$ but we avoid this to simplify the presentation.

and the analyzer only observes the sum of these vectors (the output of $\hat{\mathcal{S}}_d$). Refer to Ács and Castelluccia [5], Shi, Chan, Rieffel, Chow, and Song [55], and Kairouz, Liu, and Steinke [47] (and citations within) for example protocols.

A recurring constraint is that the trusted primitive can only perform arithmetic modulo $q$; we make this explicit by writing $\hat{\mathcal{S}}_{d,q}$. Ishai et al. show that, given messages $y_1, \ldots, y_n \in \mathbb{Z}_q$, we can essentially approximate the behavior of $\hat{\mathcal{S}}_{d,q}$ on the messages by running the shuffler $\mathcal{S}$ on an encoding of each $y_i$ [44]. Balle et al. apply this result to distributed differential privacy: any secure aggregation protocol can be simulated by a shuffle protocol that has very similar privacy guarantees [9]. These works give a bound on the communication complexity to perform this simulation that is strengthened in follow-up work by Balle et al. and Ghazi et al. [10, 39]. We summarize this line of work in the lemma below.

**Lemma 1.3.15** (Via [44, 9] and [10, 39])**.** *Let $\hat{\mathcal{P}} = (\hat{\mathcal{R}}, \hat{\mathcal{S}}_{d,q}, \hat{\mathcal{A}})$ be a secure aggregation protocol that satisfies $(\varepsilon, \delta)$ differential privacy for $n$ users. There exists a shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ that satisfies $(\varepsilon, 2\delta)$-differential privacy for $n$ users and, on any input $\vec{x}$, $\mathcal{P}(\vec{x}) = \hat{\mathcal{P}}(\vec{x})$. Each user in $\mathcal{P}$ sends $d \cdot \lceil 2 + \frac{2\log_2(d/\delta)+\log_2 q}{\log_2(n/e)} \rceil$ messages, each consisting of $\lceil \log_2 d \cdot \log_2 q \rceil$ bits.*[3]

The upshot is that private protocols the shuffle model are at least as strong as private protocols in this iteration of the secure aggregation model.

Now suppose we relax the constraint to non-negativity. Specifically, each user sends one vector $\in \mathbb{Z}^d_{\geq 0}$ to a secure aggregator $\hat{\mathcal{S}}_{d,\geq 0}$. The aggregator computes the sum of all these vectors and reports it to the analyzer without any modification. In Appendix A.2, we show that the shuffle model is equivalent to this stronger form of the secure aggregation model. The equivalence comes from the fact that a histogram that summarizes a multiset of objects (the output of aggregator $\mathcal{S}_{d,\geq 0}$) contains the same information as does a uniformly random permutation of those same objects (the output of shuffler $\mathcal{S}$).

The following is immediate from steps inside the proof:

**Lemma 1.3.16.** *Fix any shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ and $n \in \mathbb{N}$. Let $\mathcal{M}_{m,\mathcal{R}}$ denote the algorithm that, on input $x_1, \ldots, x_m$, executes $\mathcal{R}(x_1), \ldots, \mathcal{R}(x_m)$ and reports the histogram of all generated messages. If $\mathcal{M}_{\gamma n,\mathcal{R}}$ is $(\tilde{\varepsilon}(\gamma), \tilde{\delta}(\gamma))$-differentially private for all $\gamma \in [\tau, 1]$ such that $\gamma n \in \mathbb{N}$, then $\mathcal{P}$ is $(\tilde{\varepsilon}(\gamma), \tilde{\delta}(\gamma), \tau)$-robustly shuffle private for $n$ users.*

We will use this lemma to prove our protocols satisfy robust privacy.

### 1.3.3 Online Algorithms

As a means to obtain lower bounds for robustly private shuffle protocols, we take a brief detour to the online model. An algorithm in the online model receives raw data in a stream. At each step in the stream, the algorithm receives a data point, updates its internal state based on this data point, and then proceeds to the next element. The only way the algorithm "remembers" past elements is through its internal state. We formally define this below:

**Definition 1.3.17** (Online Algorithm)**.** An *online algorithm* $\mathcal{Q}$ is defined by a sequence of internal algorithms $\mathcal{Q}_1, \mathcal{Q}_2, \ldots$ and an output algorithm $\mathcal{Q}_{\mathcal{O}}$. On input $\vec{x}$, the first function $\mathcal{Q}_1 : \mathcal{X} \to \mathcal{I}$ maps $x_1$ to a state $s_1$ and the remaining functions $\mathcal{Q}_i$ map $x_i$ and the previous state $s_{i-1}$ to a new state $s_i$. At the end of the stream, $\mathcal{Q}$ publishes a final output by executing $\mathcal{Q}_{\mathcal{O}} : \mathcal{I} \to \mathcal{O}$ on its final internal state.

[3] We remark that the prior aggregation-to-shuffle transformation did not explicitly consider high-dimensional vectors. To arrive at the bounds on message length and message complexity, we have simply applied the transformation once for each coordinate. A $\log d$ term appears in the message length to account for "labels" needed to disambiguate between executions (see Section 2.2.1 for a more precise explanation).

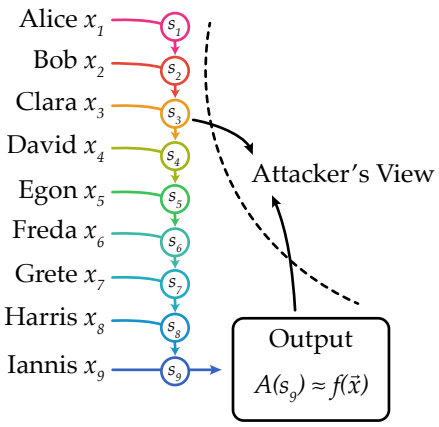


Figure 1.3: Schematic of pan-privacy.

As in the case of datasets, we say that two streams $\vec{x}$ and $\vec{x}'$ are neighbors if they differ in at most one element. Prior work established the notion of *pan-privacy* in the online model, which requires the algorithm's internal state and output to be differentially private with regard to neighboring streams.

**Definition 1.3.18** (Pan-privacy [29, 6])**.** Given an online algorithm $\mathcal{Q}$, let $\mathcal{Q}_{\mathcal{I}}(\vec{x})$ denote its internal state after processing stream $\vec{x}$, and let $\vec{x}_{\leq t}$ be the first $t$ elements of $\vec{x}$. We say $\mathcal{Q}$ is $(\varepsilon, \delta)$-*pan-private* if, for every pair of neighboring streams $\vec{x}$ and $\vec{x}'$, every time $t$ and every set of internal state, output state pairs $T \subset \mathcal{I} \times \mathcal{O}$,

$$\mathop{\mathbb{P}}_{\mathcal{Q}}\Big[\Big(\mathcal{Q}_{\mathcal{I}}(\vec{x}_{\leq t}), \mathcal{Q}_{\mathcal{O}}(\mathcal{Q}_{\mathcal{I}}(\vec{x}))\Big) \in T\Big] \leq e^{\varepsilon} \cdot \mathop{\mathbb{P}}_{\mathcal{Q}}\Big[\Big(\mathcal{Q}_{\mathcal{I}}(\vec{x}'_{\leq t}), \mathcal{Q}_{\mathcal{O}}(\mathcal{Q}_{\mathcal{I}}(\vec{x}'))\Big) \in T\Big] + \delta. \tag{1.2}$$

When $\delta = 0$, we say $\mathcal{Q}$ is $\varepsilon$-*pan-private*.

Taken together, these requirements protect against an adversary that sees any one internal state of $\mathcal{Q}$ as well as its final output. Refer to Figure 1.3 for a schematic. Our definition of pan-privacy is the specific variant given by Amin, Joseph, and Mao [6]. This version guarantees record-level (uncertainty about the presence of any single stream element) rather than user-level (uncertainty about the presence of any one data universe element) privacy. We use this variant because, like the shuffle model, we assume each data contributor has a single data point.

The maintenance of the differentially private internal state offers a stronger guarantee than in central privacy, since it protects users against future events. For example, a user may trust the current algorithm operator but want protection against the possibility that the operator will be acquired or subpoenaed in the future. Under pan-privacy, post-processing (Fact 1.3.5) ensures that future views of the pan-private algorithm's state will be differentially private with respect to past data.

We remark that lower bounds for pan-private algorithms typically only rely on the privacy of the internal state (see e.g. Theorem 12 in [51] and Theorem 3 in [6]). This means they are in fact lower bounds for a weaker notion we call *internal privacy*:

**Definition 1.3.19** (Internal Privacy)**.** An online algorithm $\mathcal{Q}$ is $(\varepsilon, \delta)$-*internally private* if, for every pair of neighboring streams $\vec{x}$ and $\vec{x}'$, every time $t$ and every set of internal states $T \subset \mathcal{I}$,

$$\mathop{\mathbb{P}}_{\mathcal{Q}}\Big[\Big(\mathcal{Q}_{\mathcal{I}}(\vec{x}_{\leq t}) \in T\Big] \leq e^{\varepsilon} \cdot \mathop{\mathbb{P}}_{\mathcal{Q}}\Big[\Big(\mathcal{Q}_{\mathcal{I}}(\vec{x}'_{\leq t})\Big) \in T\Big] + \delta. \tag{1.3}$$

When $\delta = 0$, we say $\mathcal{Q}$ is $\varepsilon$-*internally private*.

We will use internal privacy solely as a means to obtain lower bounds for robust shuffle privacy. We do not advocate its use as a design constraint, as pan-privacy protects against a more natural adversary.

# Chapter 2

# Novel Shuffle Protocols

In this chapter, we present robustly shuffle private protocols for a variety of statistical problems. As shown in Table 1.1, there are many polynomial separations in sample complexity between local privacy and robust shuffle privacy. In the special case of pointer-chasing, the separation is proportional to a parameter of the problem so that the sample complexities can be arbitrarily far apart.

## 2.1 Binary Sums

In this section, we will focus on the most basic statistical operation: each user $i$ has a bit $x_i \in \{0,1\}$ and the objective is to compute the sum up to a small amount error with high probability.

**Definition 2.1.1.** A protocol $\mathcal{P}$ *computes binary sums up to error* $\alpha$ if, for any input $\vec{x} \in \{0,1\}^n$,

$$\mathop{\mathbb{P}}_{z\sim\mathcal{P}(\vec{x})}\left[|z - \sum_{i=1}^{n} x_i| > \alpha\right] < 1/100.$$

Our claim is that the shuffle model admits protocols for binary sums with error that depends solely on privacy parameters:

**Theorem 2.1.2** (Informal)**.** *For any $\varepsilon \leq 1$, sufficiently small $\delta$, and any $n \in \mathbb{N}$, there is a shuffle protocol that is $(\varepsilon,\delta)$-robustly private for $n$ users and computes binary sums up to error $O(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}})$.*

In contrast, work by Beimel Nissim & Omri and Chan Shi & Song yields a lower bound of $\Omega(\frac{1}{\varepsilon}\sqrt{n})$ under local privacy [14, 21].

We study the binary sum protocol $\mathcal{P}_{\mathsf{SYM}} = (\mathcal{R}_{\mathsf{SYM}}, \mathcal{A}_{\mathsf{SYM}})$, adapted from joint work with Balcer, Joseph, and Mao [8]. Refer to Algorithms 1 and 2 for pseudocode. It enjoys many useful properties of the binomial, Laplace, and Gaussian mechanisms in the central model: the error in the estimate is symmetrically distributed about 0 and is independent of the input values. These properties will be useful when constructing our uniformity tester (Sec. 2.5); they are not present in the protocols $\mathcal{P}_{\mathsf{ZSUM}}$ (Sec. 2.2.2) and $\mathcal{P}_{\mathsf{RR}}$ (Sec. 4.1).

We first sketch a simple but not robust variant of the protocol. Suppose each user $i$ sends their data bit in the clear but user 1 also reports $\lambda$ bits drawn from $\mathbf{Ber}(1/2)$. Because the shuffler removes all information about the senders of the messages, the analyzer's view is *a shuffled set of $n+\lambda$ messages*. Because each message is a bit and the number of messages is public, the set of messages contains the same amount of information as the sum of the messages. This is equal to the sum of the data bits $\sum x_i$ plus the noise drawn from $\mathbf{Bin}(\lambda, 1/2)$. This is enough for differential privacy via Lemma 1.3.7. To recover an unbiased estimate, the analyzer simply subtracts $\lambda/2$ from the message sum.

The above sketch is not robust due to the fact that all privacy guarantees vanish when user 1 fails to inject the random bits. Our final protocol $\mathcal{P}_{\mathsf{SYM}}$ essentially performs randomized load-balancing: every user reports a random number of bits drawn from $\mathbf{Ber}(1/2)$. This random number of bits is sampled from $\mathbf{Pois}(\lambda/n)$. One useful property of the Poisson distribution is closure under convolution: if $\gamma n$ users execute the randomizer, the total number of noise bits is drawn from $\mathbf{Pois}(\gamma\lambda)$. We will invoke concentration of the Poisson distribution (Lemma 2.1.3 below) to give bounds on the privacy parameters as a function of $\gamma$.

**Lemma 2.1.3** (Theorem 1 in [20]). *For $X \sim \mathbf{Pois}(\lambda)$ and $t > 0$,*

$$\mathbb{P}\left[|X - \lambda| \geq t\right] \leq 2\exp\left(-\frac{t^2}{2(\lambda + t)}\right)$$

We remark that it is possible to de-randomize the number of messages: have each user send in $\lceil \lambda/n \rceil$ messages with probability 1. But to maintain the desirable symmetric noise property, the mean of each noise bit must be exactly 1/2. For $n > \lambda$, each user is sending one extra Bernoulli bit so the variance of the estimator would scale with $n$.

**Algorithm 1:** $\mathcal{R}_{\mathsf{SYM}}$, a randomizer for private binary sums

**Input:** User data $x \in \{0,1\}$
**Output:** Message vector $\vec{y} \in \{0,1\}^*$
Initialize message vector $\vec{y} \leftarrow (x)$
Sample $s \sim \mathbf{Pois}(\lambda/n)$
**For** $t \in [s]$
  $b_t \sim \mathbf{Ber}(1/2)$
  Append $b_t$ to $\vec{y}$
**Return** $\vec{y}$

**Algorithm 2:** $\mathcal{A}_{\mathsf{SYM}}$, an analyzer for private binary sums

**Input:** Message vector $\vec{y} \in \{0,1\}^*$
**Output:** Estimate $z \in \mathbb{R}$
Calculate noise scale $\ell \leftarrow |\vec{y}| - n$
Compute $z \leftarrow \left(\sum_{i=1}^{|\vec{y}|} y_i\right) - \ell/2$
**Return** $z$

Theorem 2.1.2 follows from the stronger statement below (take $\beta = 1/100$):

**Theorem 2.1.4.** *For any $\varepsilon \leq 1$, $\delta < \beta < 2e^{-9}$, and $n \in \mathbb{N}$, there is a choice of $\lambda > 0$ such that $\mathcal{P}_{\mathsf{SYM}} = (\mathcal{R}_{\mathsf{SYM}}, \mathcal{A}_{\mathsf{SYM}})$ is $(\varepsilon, \delta)$-robustly private for $n$ users and, for any input $\vec{x} \in \{0,1\}^n$,*

$$\mathbb{P}_{z \sim \mathcal{P}_{\mathsf{SYM}}(\vec{x})}\left[|z - \sum_{i=1}^{n} x_i| > \frac{11}{\varepsilon} \cdot \sqrt{\ln\frac{4}{\delta}\ln\frac{4}{\beta}}\right] < \beta$$

*Proof.* We will first bound the error of the protocol in terms of the parameter $\lambda$:

**Claim 2.1.5.** *Fix $\beta < 1$. If $\lambda > 4\ln\frac{4}{\beta}$, then for any input $\vec{x} \in \{0,1\}^n$,*

$$\mathbb{P}\left[|\mathcal{P}_{\mathsf{SYM}}(\vec{x}) - \sum x_i| < \sqrt{\lambda \ln\frac{4}{\beta}}\right] > 1 - \beta.$$

Now we derive a setting of $\lambda$ in order to achieve a target level of robust privacy:

**Claim 2.1.6.** *Fix any $\varepsilon \le 1$, $\delta < 2e^{-9}$, and $n \in \mathbb{N}$. If $\lambda \leftarrow \frac{104}{\varepsilon^2} \ln \frac{4}{\delta}$ then $\mathcal{P}_{\mathsf{SYM}}$ is $\left(\varepsilon/\sqrt{\gamma}, \delta\right)$-robustly shuffle private for $n$ users.*

We prove the two claims later on. The theorem will immediately follow by substitution of $\lambda$ into Claim 2.1.5. □

Now it remains to prove Claims 2.1.5 and 2.1.6.

*Proof of Claim 2.1.5.* Both the binomial and Poisson distribution are closed under summation:

$$\sum_{i=1}^{r} \mathbf{Bin}(a_i, p) = \mathbf{Bin}\left(\sum_{i=1}^{r} a_i, p\right) \text{ and } \sum_{i=1}^{r} \mathbf{Pois}(\lambda_i) = \mathbf{Pois}\left(\sum_{i=1}^{r} \lambda_i\right).$$

Recall also that the count of noisy messages generated by any honest user is distributed as $\mathbf{Pois}(\lambda/n)$. Thus, the sum of the messages produced by $(\mathcal{S} \circ \mathcal{R}^n_{\mathsf{SYM}})(\vec{x})$ must be distributed as

$$\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} \mathbf{Bin}\left(\mathbf{Pois}\left(\frac{\lambda}{n}\right), 1/2\right) = \sum_{i=1}^{n} x_i + \mathbf{Bin}\left(\mathbf{Pois}(\lambda), 1/2\right)$$

We note that the random variable $\ell$ computed by $\mathcal{A}_{\mathsf{SYM}}$ is precisely the sample from $\mathbf{Pois}(\lambda)$ in the above expression. Thus $\eta \leftarrow \mathcal{P}_{\mathsf{SYM}}(\vec{x}) - \sum_{i=1}^{n} x_i$ is a random variable drawn from $\mathbf{Bin}(\ell, 1/2) - \ell/2$.

From Lemma 2.1.3,

$$\begin{aligned} \mathop{\mathbb{P}}_{\ell \sim \mathbf{Pois}(\lambda)}[\ell \ge 2\lambda] &\le 2\exp\left(-\frac{\lambda^2}{2(\lambda+\lambda)}\right) \\ &= 2\exp(-\lambda/4) \\ &\le \beta/2 \end{aligned}$$

From Hoeffding's inequality,

$$\mathbb{P}\left[|\eta| > \sqrt{\frac{\ell}{2} \ln \frac{4}{\beta}}\right] < \beta/2$$

The result follows from a union bound. □

*Proof of Claim 2.1.6.* Recall Lemma 1.3.16: proving that $\mathcal{M}_{\gamma n, \mathsf{SYM}}$ is $(\varepsilon/\sqrt{\gamma}, \delta)$-differentially private for all $\gamma \ge 1/2$ will imply that $\mathcal{P}_{\mathsf{SYM}}$ is $(\varepsilon/\sqrt{\gamma}, \delta)$-robustly shuffle private for $n$ users. And recall that $\mathcal{M}_{\gamma n, \mathsf{SYM}}$ is the algorithm that, on input $x_1, \ldots, x_{\gamma n}$, outputs the histogram which counts the occurrences of $\{0, 1\}$ as produced by $\mathcal{R}_{\mathsf{SYM}}(x_1), \ldots, \mathcal{R}_{\mathsf{SYM}}(x_{\gamma n})$.

The total number of bits produced by those $\gamma n$ executions is $\gamma n + \tilde{\ell}$ where $\tilde{\ell} \sim \mathbf{Pois}(\gamma\lambda)$. This is immediate from the fact that $\mathcal{R}_{\mathsf{SYM}}$ sends the user's bit along with $\mathbf{Pois}(\lambda/n)$ bits.

Lemma 2.1.3 implies that

$$\begin{aligned} \mathop{\mathbb{P}}_{\tilde{\ell} \sim \mathbf{Pois}(\gamma\lambda)}\left[\tilde{\ell} < \gamma\lambda/2\right] &< 2\exp(-\gamma\lambda/12) \\ &= 2\exp\left(-\frac{104\gamma}{12\varepsilon^2} \cdot \ln \frac{4}{\delta}\right) \\ &\le 2\exp\left(-\ln \frac{4}{\delta}\right) && (\gamma \ge 1/2, \varepsilon \le 1) \\ &= \delta/2 && () \end{aligned}$$

The rest of the proof conditions on $\tilde{\ell} \geq \gamma\lambda/2$. Because the number of messages is $\gamma n + \tilde{\ell}$, the frequency of 0 is computable from the frequency of 1: if $h_0, h_1$ count zeroes and ones, respectively, then $h_0 = \gamma n + \tilde{\ell} - h_1$. By post-processing (Fact 1.3.5), it suffices to prove that the count of 1 as produced by $\mathcal{R}_{\mathsf{SYM}}(x_1), \ldots, \mathcal{R}_{\mathsf{SYM}}(x_{\gamma n})$ is a differentially private algorithm.

By construction of $\mathcal{R}_{\mathsf{SYM}}$, this count is distributed as $\sum x_i + \mathbf{Bin}(\tilde{\ell}, 1/2)$. Once we show that $\tilde{\ell}/2 > 16 \ln \frac{4}{\delta}$, we can invoke Lemma 1.3.7: adding noise from $\mathbf{Bin}(\tilde{\ell}, 1/2)$ to a binary sum suffices for $(\varepsilon', \delta/2)$-differential privacy, where

$$\begin{aligned} \varepsilon' &:= \sqrt{\frac{13 \ln(4/\delta)}{\tilde{\ell}/2}} \\ &\leq \sqrt{\frac{52 \ln(4/\delta)}{\gamma\lambda}} && (\tilde{\ell} \geq \gamma\lambda/2) \\ &\leq \varepsilon/\sqrt{\gamma} \end{aligned}$$

We finally lower bound $\tilde{\ell}/2$ by $13 \ln \frac{4}{\delta}$:

$$\begin{aligned} \tilde{\ell}/2 &\geq \gamma\lambda/4 && (\text{From } \tilde{\ell} \geq \gamma\lambda/2) \\ &= \frac{26\gamma}{\varepsilon^2} \cdot \ln \frac{4}{\delta} && (2.1) \\ &> 13 \ln \frac{4}{\delta} && (\gamma \geq 1/2, \varepsilon \leq 1) \end{aligned}$$

Because we execute an $(\varepsilon/\sqrt{\gamma}, \delta/2)$-private algorithm with probability $\geq 1 - \delta/2$, we have $(\varepsilon/\sqrt{\gamma}, \delta)$-differential privacy. □

The following technical claim gives some more detail on the noise introduced by $\mathcal{P}_{\mathsf{SYM}}$, which will be useful in our uniformity tester (Sec. 2.5).

**Claim 2.1.7.** *For any $\vec{x} \in \{0,1\}^n$ and choice of $\lambda > 0$, $\mathcal{P}_{\mathsf{SYM}}(\vec{x}) - \sum x_i$ is independent of $\vec{x}$ and symmetrically distributed over the set $\{\ldots, -3/2, -1, -1/2, 0, 1/2, 1, 3/2, \ldots\}$ such that the first four central moments are $0, \lambda/4, 0, 3\lambda^2/10 + 7\lambda/40$.*

*Proof.* Recall that we showed $\eta \leftarrow \mathcal{P}_{\mathsf{SYM}}(\vec{x}) - \sum_{i=1}^{n} x_i$ is a random variable drawn from $\mathbf{Bin}(\ell, 1/2) - \ell/2$. Symmetry of this distribution immediately follows from the symmetry of $\mathbf{Bin}(\ell, 1/2)$. In the case where $\ell$ is even, the support of $\mathbf{Bin}(\ell, 1/2) - \ell/2$ is $\mathbb{Z}$. In the case where $\ell$ is odd, the support is $\{\ldots, -3/2, -1/2, 1/2, 3/2, \ldots\}$.

We now focus on the moments. For *any* $\ell$, we note that the first and third central moments of $\mathbf{Bin}(\ell, 1/2)$ are both 0, while the second is $\ell/4$ and the fourth is $\frac{3\ell^2}{10} - \frac{\ell}{8}$. Given that

$$\mathbb{E}\left[\eta^k\right] = \mathop{\mathbb{E}}_{\ell \sim \mathbf{Pois}(\lambda)} \left[ \mathop{\mathbb{E}}_{\nu \sim \mathbf{Bin}(\ell, 1/2) - \ell/2} \left[ \nu^k \right] \right],$$

we have that $\mathbb{E}\left[\eta^2\right] = \lambda/4$ and

$$\begin{aligned}
\mathbb{E}\left[\eta^4\right] &= \mathop{\mathbb{E}}_{\ell\sim\mathbf{Pois}(\lambda)}\left[\frac{3\ell^2}{10} - \frac{\ell}{8}\right] \\
&= \frac{3}{10}\cdot \mathop{\mathbb{E}}_{\ell\sim\mathbf{Pois}(\lambda)}\left[\ell^2\right] - \frac{\lambda}{8} \\
&= \frac{3}{10}\cdot\left(\mathop{\mathrm{Var}}_{\ell\sim\mathbf{Pois}(\lambda)}[\ell] + \mathop{\mathbb{E}}_{\ell\sim\mathbf{Pois}(\lambda)}[\ell]^2\right) - \frac{\lambda}{8} \\
&= \frac{3}{10}\cdot\left(\lambda + \lambda^2\right) - \frac{\lambda}{8} \\
&= 3\lambda^2/10 + 7\lambda/40
\end{aligned}$$

□

### 2.1.1 Application: Mean Estimation of Distributions over [0,1]

In this subsection, we consider data distributions **D** over the continuous but bounded interval $[0,1]$. The goal is to privately estimate the mean of **D** up to some prescribed error $\alpha$. More formally,

**Definition 2.1.8** (Mean Estimation of Bounded Data)**.** Let $\alpha$ be any real in the interval $(0,1/2)$. An algorithm $\mathcal{M}$ estimates the mean of bounded data up to error $\alpha$ with sample compexity $n$ if, for any distribution **D** over $[0,1]$, it takes $n$ independent samples from **D** and reports $z$ such that $|z - \mathop{\mathbb{E}}_{X\sim\mathbf{D}}[X]| < \alpha$ with probability at least 99/100. This probability is taken over the randomness of the samples observed by $\mathcal{M}$ and the algorithm $\mathcal{M}$ itself.

We prove the following upper bound on the sample complexity under robust shuffle privacy:

**Theorem 2.1.9** (Informal)**.** *For $\varepsilon \le 1$ and sufficiently small $\delta$, there exists an $(\varepsilon,\delta)$-robustly private shuffle protocol that estimates the mean of bounded data up to error $\alpha$ with sample complexity $O(\frac{1}{\alpha^2} + \frac{1}{\alpha\varepsilon}\sqrt{\log\frac{1}{\delta}})$.*

The main idea is to combine the $\mathcal{P}_{\mathsf{SYM}}$ protocol with randomized rounding: replace user $i$'s real value $x_i$ with a binary value $b_i \in \{0,1\}$ with expected value $x_i$ and then run the protocol on these new bits.

We define the randomizer and analyzer below

$$\mathcal{R}_{\mathsf{ME}}(x \in \{0,1\}) := \mathcal{R}_{\mathsf{SYM}}(b \sim \mathbf{Ber}(x)) \tag{2.2}$$

$$\mathcal{A}_{\mathsf{ME}}(\vec{y} \in \{0,1\}^*) := \frac{1}{n}\cdot\mathcal{A}_{\mathsf{SYM}}(\vec{y}) \tag{2.3}$$

Because there are three sources of error—sampling, rounding, and privacy—we give sample complexity guarantees for each. The following bound on the non-private sample complexity is immediate from an additive Chernoff bound:

**Claim 2.1.10.** *Fix $n > \frac{1}{\alpha^2}\ln\frac{2}{\beta}$. If we sample $\vec{x}\sim\mathbf{D}^n$ where **D** is a distribution over $[0,1]$ and $\mathop{\mathbb{E}}_{x\sim\mathbf{D}}[x] = \mu$, then*

$$\mathbb{P}\left[|\mu - \frac{1}{n}\sum_{i=1}^{n} x_i| < \alpha\right] \ge 1-\beta$$

We can also obtain the following characterization of the randomized rounding:

**Claim 2.1.11.** *Fix $n > \frac{1}{\alpha^2}\ln\frac{2}{\beta}$ and any $\vec{x}\in[0,1]^n$. If $b_i \sim \mathbf{Ber}(x_i)$, then*

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n} x_i - \frac{1}{n}\sum_{i=1}^{n} b_i\right| < \alpha\right] \ge 1-\beta$$

Finally, the following is a rephrasing of Theorem 2.1.4:

**Corollary 2.1.12.** *For any $\varepsilon \le 1$, $\delta < \beta < 2e^{-9}$, and $n = \Omega(\frac{1}{\alpha\varepsilon}\sqrt{\log\frac{1}{\delta}\log\frac{1}{\beta}})$, there exists a choice of parameter $\lambda$ such that $\mathcal{P}_{\mathsf{SYM}} = (\mathcal{R}_{\mathsf{SYM}}, \mathcal{A}_{\mathsf{SYM}})$ satisfies $(\varepsilon,\delta)$-robust shuffle privacy and, for all $\vec{b} \in \{0,1\}^n$,*

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n} b_i - \frac{1}{n}\cdot\mathcal{P}_{\mathsf{SYM}}(\vec{b})\right| < \alpha\right] \ge 1-\beta$$

Theorem 2.1.9 is immediate from a union bound over Claim 2.1.10, Claim 2.1.11 and Corollary 2.1.12.

### 2.1.2 Application: Feature Selection

In this subsection, we consider distributions **D** over the boolean hypercube $\{0,1\}^d$ (equivalently, $\{\pm1\}^d$). The goal is to privately identify a coordinate with the largest mean. More formally,

**Definition 2.1.13** (Feature Selection Problem)**.** Let $\alpha$ be any real in the interval $(0,1/2)$ and let $d$ be any integer larger than 1. An algorithm $\mathcal{M}$ *solves $(\alpha,d)$-feature selection with sample complexity $n$* if, for any distribution **D** over $\{0,1\}^d$, it takes $n$ independent samples from **D** and selects a coordinate $J \in [d]$ such that $\mathop{\mathbb{E}}_{X\sim\mathbf{D}}\left[X_J\right] \ge \max_j \mathop{\mathbb{E}}_{X\sim\mathbf{D}}\left[X_j\right] - \alpha$ with probability at least 99/100. This probability is taken over the randomness of the samples observed by $\mathcal{M}$ and the algorithm $\mathcal{M}$ itself.

**Theorem 2.1.14** (Informal)**.** *For $\varepsilon \le 1$ and sufficiently small $\delta$, there exists an $(\varepsilon,\delta)$-robustly shuffle private protocol that solves $(\alpha,d)$-feature selection with sample complexity $n = \tilde{O}\left(\frac{\log d}{\alpha^2} + \frac{\sqrt{d}}{\alpha\varepsilon}\log\frac{1}{\delta}\right)$.*

Our solution is $\mathcal{P}_{\mathsf{FS}} = (\mathcal{R}_{\mathsf{FS}}, \mathcal{A}_{\mathsf{FS}})$ whose pseudocode is given in Algorithms 3 and 4. The approach we take is to estimate the mean of each coordinate by executing $\mathcal{P}_{\mathsf{SYM}}$ multiple times (essentially adapting $\mathcal{P}_{\mathsf{ME}}$ from Section 2.1.1 to a multi-dimensional setting). Note that the $d$ executions are done in one round of communication. This is achieved by labeling the messages by their execution number and then packaging all messages together. More precisely, if $y_{i,1}^{(j)}, y_{i,2}^{(j)}, \dots$ denote the messages user $i$ wishes to transmit in execution $j$, they can send the vector $(\dots,(j,y_{i,1}^{(j)}),(j,y_{i,2}^{(j)}),\dots)$. The labeling allows the analyzer to disambiguate the executions of $\mathcal{P}_{\mathsf{SYM}}$. The labeling technique was first presented in joint work with Smith, Ullman, Zeber, and Zhilyaev [23].

**Algorithm 3:** $\mathcal{R}_{\mathsf{FS}}$, a randomizer for private feature selection

**Input:** $x \in \{0,1\}^d$; implicit parameter $\lambda > 0$
**Output:** $\vec{y} \in ([d]\times\{0,1\})^*$
Initialize message vector $\vec{y} \leftarrow \emptyset$
**For** $j \in [d]$
  Compute $\vec{y}^{(j)} \leftarrow \mathcal{R}_{\mathsf{SYM}}(x_j)$ using parameter $\lambda$
  **For** $y \in \vec{y}^{(j)}$
    Append tuple $(j,y)$ to $\vec{y}$
**Return** $\vec{y}$

Because there are two sources of error—sampling and privacy—we give sample complexity guarantees for each. The following bound on the non-private sample complexity is immediate from applying a union bound to Claim 2.1.10:

**Algorithm 4:** $\mathcal{A}_{\mathsf{FS}}$, an analyzer for private feature selection

**Input:** $\vec{y} \in ([d] \times \{0,1\})^*$; implicit parameter $\lambda > 0$
**Output:** $j \in [d]$
**For** $j \in [d]$
  Let $\vec{y}^{(j)}$ be those bits labeled by $j$ in input $\vec{y}$
  $z_j \leftarrow \frac{1}{n} \cdot \mathcal{A}_{\mathsf{SYM}}(\vec{y}^{(j)})$
**Return** $\arg\max_{j \in [d]} z_j$

**Corollary 2.1.15.** *Fix $n > \frac{1}{\alpha^2} \ln \frac{2d}{\beta}$. If we sample $\vec{x} \sim \mathbf{D}^n$ where $\mathbf{D}$ is a distribution over $\{0,1\}^d$ and $\mathop{\mathbb{E}}_{x \sim \mathbf{D}}\left[x_j\right] = \mu_j$, then*

$$\mathop{\mathbb{P}}_{\vec{x} \sim \mathbf{D}^n}\left[\forall j \ \left|\mu_j - \frac{1}{n}\sum_{i=1}^{n} x_{i,j}\right| < \alpha\right] \geq 1 - \beta.$$

Now we consider the error due to privacy:

**Claim 2.1.16.** *For any $\varepsilon \leq 1$, $\delta < \beta < 2e^{-9}$, and $n = \tilde{\Omega}\left(\frac{\sqrt{d}}{\alpha\varepsilon} \log \frac{1}{\delta} \sqrt{\log \frac{1}{\beta}}\right)$, there exists a choice of parameter $\lambda$ such that $\mathcal{P}_{\mathsf{FS}} = (\mathcal{R}_{\mathsf{FS}}, \mathcal{A}_{\mathsf{FS}})$ satisfies $(\varepsilon, \delta)$-robust shuffle privacy and, for all $\vec{x} \in (\{0,1\}^d)^n$ and $z_j$ computed by $\mathcal{P}_{\mathsf{FS}}(\vec{x})$,*

$$\mathbb{P}\left[\forall j \ \left|\frac{1}{n}\sum_{i=1}^{n} x_{i,j} - z_j\right| < \alpha\right] \geq 1 - \beta$$

*Proof.* Our first objective will be to prove the privacy claim. Define $\varepsilon' := \varepsilon \cdot \frac{1}{\sqrt{8d \ln(1/\delta)}}$ and $\delta' := \delta/(d+1)$; note that $\varepsilon' < 1$. Choose $\lambda$ such that $\mathcal{P}_{\mathsf{SYM}}$ satisfies $(\varepsilon', \delta')$-robust shuffle privacy. By advanced composition (Fact 1.3.4), $\mathcal{P}_{\mathsf{FS}}$ satisfies $\left(\varepsilon' \cdot \sqrt{2d \ln(1/\delta)} + \varepsilon' \cdot (\exp(\varepsilon') - 1) \cdot d, \delta' \cdot (d+1)\right)$-robust shuffle privacy. Note that the second parameter is $\delta$ by substitution. We can simplify the first parameter:

$$\begin{aligned}
&\varepsilon' \cdot \sqrt{2d \ln(1/\delta)} + \varepsilon' \cdot (\exp(\varepsilon') - 1) \cdot d \\
&= \frac{\varepsilon}{2} + \varepsilon' \cdot (\exp(\varepsilon') - 1) \cdot d \\
&\leq \frac{\varepsilon}{2} + 2(\varepsilon')^2 \cdot d \\
&= \frac{\varepsilon}{2} + \varepsilon^2 \cdot \frac{1}{4 \ln(1/\delta)} \\
&\leq \varepsilon
\end{aligned}$$

The inequalities come from $\varepsilon \leq 1$.

Observe that each $z_j$ is an estimate of the binary sum $\frac{1}{n}\sum_{i=1}^{n} x_{i,j}$ as produced by $\mathcal{P}_{\mathsf{SYM}}$. By applying Theorem 2.1.4, we have that

$$\mathbb{P}\left[\forall j \ \left|\frac{1}{n}\sum_{i=1}^{n} x_{i,j} - z_j\right| = O\left(\frac{1}{\varepsilon' n}\sqrt{\log \frac{1}{\delta'} \log \frac{d}{\beta}}\right)\right] \geq 1 - \beta.$$

Substitution gives our accuracy claim. □

Theorem 2.1.14 follows from a union bound over the two preceding claims.

### 2.1.3 Other Binary Sum Protocols

There are other differentially private protocols for the binary sum problem in the shuffle model literature; Table 2.1 summarizes their message complexity and bound on error.

Table 2.1: Shuffle protocols for binary sums from across the literature. Except for [9], each message in these protocols consumes only one bit. Note that [9, 37] do not analyze their protocol under robust variant of shuffle privacy.

| Source | Error | Messages per User | Properties | Assumptions |
|---|---|---|---|---|
| [8] (Thm. 2.1.2) | $O(\frac{1}{\varepsilon}\cdot\sqrt{\log\frac{1}{\delta}})$ | $1+O(\frac{1}{\varepsilon^2 n}\log\frac{1}{\delta})$ in expectation | Noise is symmetric & data-independent | |
| [9] | $O(\frac{1}{\varepsilon})$ | $O(\log(n/\delta))$ | Optimal Error | |
| [7] (Thm. 2.2.4) | $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ | 2 | If sum is 0, estimate is 0 | |
| [37] | $O(\frac{1}{\varepsilon^{3/2}}\cdot\sqrt{\log\frac{1}{\varepsilon}})$ | $O(\frac{1}{\varepsilon}\log n)$ | $\delta = 0$ | $n > \frac{1}{\varepsilon^{3/2}}$ |
| [23] (Thm. 4.1.2) | $O(\frac{1}{\varepsilon}\cdot\sqrt{\log\frac{1}{\delta}})$ | 1 | Single-message | $n = \Omega(\frac{1}{\varepsilon}\log\frac{1}{\delta})$ |

## 2.2 Histograms

In this section, we focus on the problem of computing histograms. For any $j \in [d]$, we define the function $c_j : [d]^n \to \mathbb{R}$ as the count of $j$ in dataset $\vec{x}$:

$$c_j(\vec{x}) = \sum_{i=1}^{n} \mathbb{1}[x_i = j] \tag{2.4}$$

The histogram of $\vec{x}$ is the vector $c(\vec{x}) := (c_1(\vec{x}), \ldots, c_d(\vec{x}))$. To measure the accuracy of a histogram protocol, we will measure the maximum error of any frequency estimate. Specifically,

**Definition 2.2.1.** A protocol $\mathcal{P}$ *computes histograms up to maximum error* $\alpha$ if, for any input $\vec{x} \in [d]^n$,

$$\mathop{\mathbb{P}}_{\vec{z}\sim\mathcal{P}(\vec{x})}\left[\|\vec{z} - c(\vec{x})\|_\infty > \alpha\right] < 1/100.$$

From joint work with Balcer [7], there is a robustly shuffle private protocol where the maximum error depends on $\varepsilon$ and $\delta$ but not $d$.

**Theorem 2.2.2** (Informal)**.** *For $\varepsilon \le 1$, $\delta < 1/100n$, and sufficiently large $n$, there is a $(\varepsilon,\delta)$-robustly shuffle private protocol that computes histograms up to maximum error $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$.*

In contrast, Bassily and Smith prove that a dependence on $d$ is necessary under local privacy [12].

### 2.2.1 A protocol template $\mathcal{P}_{\mathsf{REP}}$

To build up to Theorem 2.2.2, we start by sketching a design template originally presented in joint work with Smith, Ullman, Zeber, and Zhilyaev [23]. Algorithms 5 and 6 contain the pseudocode for protocol $\mathcal{P}_{\mathsf{REP}} = (\mathcal{R}_{\mathsf{REP}}, \mathcal{A}_{\mathsf{REP}})$. It requires access to a black-box protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$. Each user repeatedly executes the randomizer $\mathcal{R}$, where the $j$-th execution is dedicated to privately estimating $c_j(\vec{x})$. To compute the histogram, the analyzer executes $\mathcal{A}$ on each batch of messages. As was done in the feature selection protocol, the $d$ executions of $\mathcal{R}$ are performed in parallel and we disambiguate by labeling the messages in each batch.

**Algorithm 5:** $\mathcal{R}_{\mathsf{REP}}$, a randomizer that repeatedly executes another randomizer

**Input:** Data $x \in [d]$ and a randomizer $\mathcal{R} : \{0,1\} \to \mathcal{Y}^*$
**Output:** $\vec{y} \in ([d] \times \mathcal{Y})^*$
Initialize $\vec{y} \leftarrow ()$
**For** $j \in [d]$
  Let $b_j \leftarrow \mathbb{1}[x = j]$
  For each message $y$ produced by $\mathcal{R}(b_j)$, append $(j, y)$ to $\vec{y}$
**Return** $\vec{y}$

**Algorithm 6:** $\mathcal{A}_{\mathsf{REP}}$, an analyzer that repeatedly executes another analyzer

**Input:** Message vector $\vec{y} \in ([d] \times \mathcal{Y})^*$ and an analyzer $\mathcal{A} : \mathcal{Y}^* \to \mathcal{Z}$
**Output:** $\vec{z} \in \mathcal{Z}^d$
**For** $j \in [d]$
  Initialize $\vec{y}_{(j)} \leftarrow ()$
  **For** $(i, y) \in \vec{y}$
    **If** $i = j$ **:**
      Append $y$ to $\vec{y}_{(j)}$
  Compute $z_j \leftarrow \mathcal{A}(\vec{y}_{(j)})$
**Return** $(z_1, \ldots, z_d)$

Unlike feature selection (Sec. 3.3), privacy composes over only two protocol executions. This is because changing a user's value from $j$ to $j'$ affects the counts of $j, j'$. We formalize this in the following lemma:

**Lemma 2.2.3.** *If the protocol $\mathcal{P}_{\mathsf{REP}} = (\mathcal{R}_{\mathsf{REP}}, \mathcal{A}_{\mathsf{REP}})$ is given access to an $(\tilde{\varepsilon}, \tilde{\delta})$-robustly shuffle private protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$, then $\mathcal{P}_{\mathsf{REP}}$ is $(2\tilde{\varepsilon}, 2\tilde{\delta})$-robustly shuffle private.*

*Proof.* We assume without loss of generality that $[\gamma n]$ are the honest users. Fix any neighboring pair of datasets $\vec{x} \sim \vec{x}' \in [d]^{\gamma n}$ for the honest users. For any $j \in [d]$, let $b_{i,(j)}$ (resp. $b'_{i,(j)}$ denote the indicator bit $\mathbb{1}[x_i = j]$ (resp. $\mathbb{1}\left[x'_i = j\right]$). Notice that $b_{i,(j)} = b'_{i,(j)}$ except when $x_i = j$ and $x'_i \neq j$. Because $\vec{x} \sim \vec{x}'$, we conclude there are precisely two indices $j, j'$ where $b_{i,(j)} \neq b'_{i,(j)}$.

Note that the set of messages labeled by $j$ in the output of $(\mathcal{S} \circ \mathcal{R}^{\gamma n}_{\mathsf{REP}})(\vec{x})$ has the same distribution as the set of messages produced by $(\mathcal{S} \circ \mathcal{R}^{\gamma n})(\vec{b}_{(j)})$. Moreover, the independence with which $\mathcal{R}$ is executed implies that this equivalence is true for all $j$ simulatenously: $(\mathcal{S} \circ \mathcal{R}^{\gamma n}_{\mathsf{REP}})(\vec{x})$ is a post-processing of $(\mathcal{S} \circ \mathcal{R}^{\gamma n})(\vec{b}_{(1)}), \ldots, (\mathcal{S} \circ \mathcal{R}^{\gamma n})(\vec{b}_{(d)})$. Because we have shown that there are precisely two indices $j, j'$ where $b_{i,(j)} \neq b'_{i,(j)}$, robust shuffle privacy follows by composition (Fact 1.3.3). □

One way to fill out the template is to use $\mathcal{P}_{\mathsf{SYM}}$ (Sec. 2.1). Theorem 2.1.4 and a union bound implies a maximum error of $O(\frac{1}{\varepsilon}\sqrt{\log(1/\delta)\log(d/\beta)})$ with probability $\geq 1 - \beta$.

Joint work with Balcer [7] gives a critical enhancement of the technique: use a specially crafted binary sum protocol $\mathcal{P}_{\mathsf{ZSUM}}$. This protocol deterministically outputs 0 if the true sum $\sum x_i$ is 0 and otherwise outputs a noisy estimate. The maximum error of the modified histogram protocol is now the maximum noise introduced to the *nonzero* counts. Since there are at most $n$ such counts, we perform a union bound over $n$ instead of $d$ executions.

### 2.2.2 The binary sum protocol $\mathcal{P}_{\mathsf{ZSUM}}$

In this subsection, we describe $\mathcal{P}_{\mathsf{ZSUM}} = (\mathcal{R}_{\mathsf{ZSUM}}, \mathcal{A}_{\mathsf{ZSUM}})$. The randomizer is given by Algorithm 7 and the analyzer by Algorithm 8.

To understand the protocol, first recall how $\mathcal{P}_{\mathsf{SYM}}$ ensures privacy: each honest user generates a random number of samples from $\mathbf{Ber}(1/2)$ so that the burden of generating binomial noise is evenly distributed. Now, consider the following modification: given a parameter $p$, each honest user deterministically samples a single bit from $\mathbf{Ber}(p)$. Although the resulting noise distribution is not symmetric ($p \neq 1/2$), note that it is bounded by $n$ with probability 1. Via truncation, this means the analyzer can ensure that the all-zeroes input has no error with probability 1.

One technical issue is that the privacy analysis will rely on an assumption that $n$ is sufficiently large. We overcome this hurdle by switching to "silent mode" for smaller $n$: here, users report nothing to the analyzer. While the error in this case will be $\leq n$, $n$ is already ensured to be small.

**Algorithm 7:** $\mathcal{R}_{\mathsf{ZSUM}}$ a randomizer for binary sums

**Input:** $x \in \{0,1\}$; parameters $p, r$
**Output:** $\vec{y} \in \{(), (1), (1,1)\}$
**If** $r = 1$ **:**
  Sample $w \sim \mathbf{Ber}(p)$.
  **Return** *the vector containing $x + w$ messages, each with value 1*
**Else**
  **Return** ()

**Algorithm 8:** $\mathcal{A}_{\mathsf{ZSUM}}$ an analyzer for binary sums

**Input:** $\vec{y} \in \{1\}^*$; parameter $p$
**Output:** $z \in \mathbb{R}$, an estimate of the binary sum
Let $\ell$ be the length of $\vec{y}$.
**If** $\ell \leq n$ **:**
  **Return** *0*
**Else**
  **Return** $\ell - n \cdot p$

**Theorem 2.2.4.** *Fix $\varepsilon \leq 1$ and $\delta \leq 2e^{-9}$. For any $n \in \mathbb{N}$, there exists choices of $p, r$ such that the protocol $\mathcal{P}_{\mathsf{ZSUM}} = (\mathcal{R}_{\mathsf{ZSUM}}, \mathcal{A}_{\mathsf{ZSUM}})$ has the following properties:*

*i. Each user sends at most two one-bit messages.*

*ii. On input $(\underbrace{0, \ldots, 0}_{n \text{ copies}})$, the protocol reports 0 with probability 1.*

*iii. $\mathcal{P}_{\mathsf{ZSUM}}$ estimates binary sums with error at most*

$$O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$$

*with probability $\geq 1 - \delta$.*

*iv. $\mathcal{P}^{\mathsf{ZSUM}}_{\varepsilon,\delta}$ is $(\varepsilon, \delta)$-robustly shuffle private for $n$ users.*

*Proof.* The first two items in the theorem statement are straightforward to verify. Because $x$ and $w$ are both bits, $x + w \leq 2$ with probability 1: no user ever sends more than two messages. And when $\vec{x} = (0, \ldots, 0)$, $\ell \leftarrow |\vec{y}|$ is either 0 or drawn from $0 + \mathbf{Bin}(n, p)$. Thus, $\ell \leq n$ with probability 1 so that $\mathcal{P}_{\mathsf{ZSUM}}(\vec{x}) = 0$.

To prove the last two items, we will employ case analysis. In the case where $n \leq \frac{52}{\varepsilon^2} \cdot \ln \frac{2}{\delta}$, we will set $r \leftarrow 0$ and $p$ to an arbitrary value. Notice that $\mathcal{R}_{\mathsf{ZSUM}}(0) = \mathcal{R}_{\mathsf{ZSUM}}(1)$ so $\mathcal{R}_{\mathsf{ZSUM}}$ satisfies $(0, 0)$-local privacy. In terms of accuracy, observe that the output of the protocol on any input is 0 with probability 1 (since $\ell = 0$). The error of the estimate is therefore at most $n = O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$ with probability 1.

In the other case, we set $r \leftarrow 1$. We will rely on the following two claims about the accuracy and privacy of the protocol. Much like $\mathcal{P}_{\mathsf{SYM}}$, we will state accuracy results in terms of $p$ and then determine the correct choice of $p$ for target shuffle privacy.

**Claim 2.2.5** (Accuracy of $\mathcal{P}_{\mathsf{ZSUM}}$)**.** *For any $\delta \geq 2\exp(-np(1-p))$ and any input $\vec{x} \in \{0,1\}^n$, the protocol $\mathcal{P}_{\mathsf{ZSUM}}$ estimates $\sum x_i$ to within $n(1-p) + 2 \cdot \sqrt{np(1-p) \cdot \ln(2/\delta)}$ with probability $\geq 1 - \delta$.*

**Claim 2.2.6** (Robust Shuffle Privacy of $\mathcal{P}_{\mathsf{ZSUM}}$)**.** *For any $\varepsilon \leq 1$, $\delta < 2e^{-9}$ if $n > \frac{52}{\varepsilon^2} \cdot \ln \frac{2}{\delta}$ and we assign $p \leftarrow 1 - \frac{26}{\varepsilon^2 n} \cdot \ln \frac{2}{\delta}$, then the protocol $\mathcal{P}_{\mathsf{ZSUM}}$ satisfies $(\varepsilon/\sqrt{\gamma}, \delta, \tau)$-robust shuffle privacy for $n$ users.*

We note that the choice of $p$ in Claim 2.2.6 implies $\delta \geq 2\exp(-np(1-p))$, as demanded by Claim 2.2.5. So we can conclude that the protocol is $(\varepsilon, \delta)$-robustly shuffle private and that following bounds the error with probability $\geq 1 - \delta$:

$$\begin{aligned}
&n(1-p) + 2 \cdot \sqrt{np(1-p) \cdot \ln \frac{2}{\delta}} \\
\leq\ &n(1-p) + 2 \cdot \sqrt{n(1-p) \cdot \ln \frac{2}{\delta}} \\
=\ &\frac{26}{\varepsilon^2} \cdot \ln \frac{2}{\delta} + 2 \cdot \sqrt{\frac{26}{\varepsilon^2} \cdot \ln \frac{2}{\delta} \cdot \ln \frac{2}{\delta}} \\
=\ &O\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta}\right)
\end{aligned}$$

The final step follows from our bound on $\varepsilon$. □

Now it simply remains to prove the two claims.

*Proof of Claim 2.2.5.* For shorthand, we define $\alpha' = 2 \cdot \sqrt{np(1-p) \cdot \ln(2/\delta)}$ and $\alpha := n(1-p) + \alpha'$. Our objective is to show that $|\mathcal{P}_{\mathsf{ZSUM}}(\vec{x}) - \sum x_i| \leq \alpha$ with probability $\geq 1 - \delta$. If we let $w_i$ be the random bit generated by the $i$-th user, note that $\sum w_i$ is drawn from $\mathbf{Bin}(n, p)$. An additive Chernoff bound implies that for our $\delta$ regime, the following event occurs with probability $\geq 1 - \delta$:

$$\left| \sum_{i=1}^{n} w_i - np \right| \leq \alpha' \tag{2.5}$$

The remainder of the proof will condition on (2.5). In the case where $\ell > n$, the analyzer outputs $\ell - np$. We show that the error of $\ell - np$ is at most $\alpha'$:

$$\begin{aligned}
\left| (\ell - np) - \sum_{i=1}^{n} x_i \right| &= \left| \sum_{i=1}^{n} (x_i + w_i) - np - \sum_{i=1}^{n} x_i \right| && \text{(By construction)} \\
&= \left| \sum_{i=1}^{n} w_i - np \right| \\
&\leq \alpha' && \text{(By (2.5))}
\end{aligned}$$

In the case where $\ell \le n$, the analyzer will output 0. This means the error is exactly $\sum x_i$. We argue that $\ell \le n$ implies $\sum x_i \le \alpha$.

$$\begin{aligned} n &\ge \ell \\ &= \sum_{i=1}^{n} (x_i + w_i) && \text{(By construction)} \\ &\ge \sum_{i=1}^{n} x_i + np - \alpha' && \text{(By (2.5))} \end{aligned}$$

Rearranging terms yields

$$\sum_{i=1}^{n} x_i \le n(1-p) + \alpha' = \alpha$$

which concludes the proof. □

*Proof of Claim 2.2.6.* Note that $1 - p < 1/2$ because $n > \frac{52}{\varepsilon^2} \ln \frac{2}{\delta}$. Hence, $1 - p = \min(p, 1-p)$.

Recall Lemma 1.3.16: proving that $\mathcal{M}_{\gamma n, \mathsf{ZSUM}}$ is $(\varepsilon/\sqrt{\gamma}, \delta)$-differentially private for all $\gamma \ge 1/2$ will imply that $\mathcal{P}_{\mathsf{SYM}}$ is $(\varepsilon/\sqrt{\gamma}, \delta)$-robustly shuffle private for $n$ users. And recall that $\mathcal{M}_{\gamma n, \mathsf{ZSUM}}$ is the algorithm that, on input $x_1, \ldots, x_{\gamma n}$, outputs the histogram $\vec{h} \in \mathbb{Z}_{\ge 0}$ which counts the number of messages produced by $\mathcal{R}_{\mathsf{SYM}}(x_1), \ldots, \mathcal{R}_{\mathsf{SYM}}(x_{\gamma n})$.

But this number is clearly distributed as $\sum_{i=1}^{\gamma n} x_i + \mathbf{Bin}(\gamma n, p)$. If we can show that $\gamma n \min(p, 1-p) \ge 13 \ln \frac{2}{\delta}$, then Lemma 1.3.7 will imply that the mechanism is $(\tilde{\varepsilon}(\gamma n, p), \delta)$-private, where

$$\begin{aligned} \tilde{\varepsilon}(\gamma n, p) := \sqrt{\frac{13}{\gamma n \min(p, 1-p)} \ln \frac{2}{\delta}} &= \sqrt{\frac{13}{\gamma n (1-p)} \ln \frac{2}{\delta}} \\ &= \sqrt{\frac{13}{26\gamma}} \cdot \varepsilon \\ &\le \varepsilon/\sqrt{\gamma} \end{aligned}$$

It remains to prove $\gamma n \min(p, 1-p) \ge 13 \ln \frac{2}{\delta}$.

$$\begin{aligned} \gamma n \min(p, 1-p) = \gamma n (1-p) &= \frac{26\gamma}{\varepsilon^2} \cdot \ln \tfrac{2}{\delta} \\ &\ge 13 \ln \frac{2}{\delta} \end{aligned}$$

The final inequality comes from $\varepsilon \le 1, \gamma \ge 1/2$ □

### 2.2.3 Filling the $\mathcal{P}_{\mathsf{REP}}$ template with $\mathcal{P}_{\mathsf{ZSUM}}$

In this section, we focus on the protocol $\mathcal{P}_{\mathsf{HIST}} := (\mathcal{R}_{\mathsf{HIST}}, \mathcal{A}_{\mathsf{HIST}})$ whose randomizer and analyzer are specified below:

$$\begin{aligned} \mathcal{R}_{\mathsf{HIST}}(\cdot) &:= \mathcal{R}_{\mathsf{REP}}(\cdot, \mathcal{R}_{\mathsf{ZSUM}}) \\ \mathcal{A}_{\mathsf{HIST}}(\cdot) &:= \mathcal{A}_{\mathsf{REP}}(\cdot, \mathcal{A}_{\mathsf{ZSUM}}) \end{aligned}$$

Theorem 2.2.2 follows from the result below concerning $\mathcal{P}_{\mathsf{HIST}}$:

**Theorem 2.2.7.** *Fix any $\varepsilon \le 1$, $n \in \mathbb{N}$, and $\delta < 2e^{-9}/n$. There are choices for parameters $p, r$ such that $\mathcal{P}_{\mathsf{HIST}}$ has the following properties:*

*i. Each user sends $\le d+1$ messages, each consisting of $O(\log d)$ bits.*

*ii. $\mathcal{P}_{\mathsf{HIST}}$ is $(2\varepsilon, 2\delta)$-robustly shuffle private for $n$ users.*

*iii. On any input $\vec{x} \in [d]^n$, $\mathcal{P}_{\mathsf{HIST}}$ reports $\vec{z}$ such that*

$$\|\vec{z} - c(\vec{x})\|_\infty = O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$$

*with probability $\ge 1 - n\delta$.*

Before proving the theorem, we remark that in Appendix 2.2.4, we show how to use hashing to reduce the message complexity from $\Theta(d)$ to $d^c$ for arbitrary constant $0 < c < 1$. Also note that the accuracy guaranteed by this protocol is close to what is possible in the central model: there is a stability-based algorithm with simultaneous error $O((1/\varepsilon) \cdot \ln(1/\delta))$ [18].

*Proof.* Part *i* is immediate from (1) the fact that each execution of $\mathcal{R}_{\mathsf{ZSUM}}$ results in at most one noise bit and (2) each user has a single value in $[d]$. Part *ii* follows from Lemma 2.2.3. Thus, it only remains to prove Part *iii*.

For any $i \in [n]$ and $j \in [d]$, let $b_{i,(j)}$ denote the bit that is 1 when $x_i = j$ (0 otherwise) and let $\vec{b}_{(j)}$ denote the vector $(b_{1,(j)}, \ldots, b_{n,(j)})$. In the output of $(\mathcal{S} \circ \mathcal{R}^n_{\mathsf{HIST}})(\vec{x})$, notice that the number of messages labeled by $j$ has the same distribution as the length of $(\mathcal{S} \circ \mathcal{R}^n_{\mathsf{ZSUM}})(\vec{b}_{(j)})$. This means $z_j$ as computed by $\mathcal{A}_{\mathsf{REP}}$ has the same distribution as $\mathcal{P}_{\mathsf{ZSUM}}(\vec{b}_{(j)})$. Thus, Claim 2.2.5 implies that $\mathbb{P}\left[|z_j - c_j(\vec{x})| > \alpha\right] < \delta$ where $\alpha = O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$.

Define $Q := \{j \in [d] : c_j(\vec{x}) > 0\}$. To bound the error, we leverage the property that when $j \notin Q$, ZSUM will report a nonzero value with probability 0.

$$\begin{aligned}
&\mathbb{P}\left[\|\vec{z} - c(\vec{x})\|_\infty > \alpha\right] \\
&= \mathbb{P}\left[\exists j \in [d] \text{ s.t. } |z_j - c_j(\vec{x})| > \alpha\right] \\
&\le \mathbb{P}\left[\exists j \in Q \text{ s.t. } |z_j - c_j(\vec{x})| > \alpha\right] + \mathbb{P}\left[\exists j \notin Q \text{ s.t. } |z_j - c_j(\vec{x})| > \alpha\right] \\
&= \mathbb{P}\left[\exists j \in Q \text{ s.t. } |z_j - c_j(\vec{x})| > \alpha\right] && \text{(Theorem 2.2.4 Part } ii\text{)} \\
&\le \sum_{j \in Q} \mathbb{P}\left[|z_j - c_j(\vec{x})| > \alpha\right] \\
&\le \sum_{j \in Q} \delta \\
&\le n\delta
\end{aligned}$$

The final inequality comes from the fact that the number of distinct elements in $\vec{x}$ is bounded by $n$. □

### 2.2.4 Reducing Message Complexity via Count-Min

Recall that the protocol $\mathcal{P}_{\mathsf{HIST}}$ has maximum error $O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$ but has message complexity $O(d)$. Using the count-min technique from the sketching literature, we can drive down the message complexity to $O(n \cdot d^{1/T})$ for arbitrary constant $T \in \mathbb{N}$ without inflating the asymptotic error. This result comes from discussion with Kobbi Nissim and Rasmus Pagh.

At a high level, we hash the universe to the domain $[\hat{d}]$. If an element $j$ experiences no collisions, the estimate produced by running $\mathcal{P}_{\mathsf{HIST}}$ on the hashed data is unaffected. Otherwise, the estimate is an

**Algorithm 9:** $\mathcal{R}_{\mathsf{HIST2}}$ a local randomizer for histograms

**Input:** $x \in [d]$; parameters $p, r \in (0,1)$, $T, \hat{d} \in \mathbb{N}$
**Output:** $\vec{y} \in ([T] \times ([\hat{d}] \times \{1\}))^*$
Obtain hash functions $\{h^{(t)} : [d] \to [\hat{d}]\}$ from public randomness.
Initialize $\vec{y} \leftarrow \emptyset$
**For** $t \in [T]$
  Compute $\vec{y}^{(t)} \leftarrow \mathcal{R}_{\mathsf{HIST}}(h^{(t)}(x))$ (using parameters $p, r$ and universe $\hat{d}$)
  **For** $y \in \vec{y}^{(t)}$
    Append $(t, y)$ to $\vec{y}$
**Return** $\vec{y}$

**Algorithm 10:** $\mathcal{A}_{\mathsf{HIST2}}$ an analyzer for histograms

**Input:** $\vec{y} \in ([T] \times ([\hat{d}] \times \{1\}))^*$; parameters $p, r \in (0,1)$, $T, \hat{d} \in \mathbb{N}$
**Output:** $\vec{z} \in \mathbb{R}^d$
Obtain hash functions $\{h^{(t)} : [d] \to [\hat{d}]\}$ from public randomness.
**For** $j \in [d]$
  $z_j \leftarrow \infty$
**For** $t \in [T]$
  Initialize $\vec{y}^{(t)} \leftarrow \emptyset$
  **For** $(t', y) \in \vec{y}$
    Append $y$ to $\vec{y}^{(t)}$ if $t' = t$
  Compute $\hat{z}^{(t)} \leftarrow \mathcal{A}_{\mathsf{HIST}}(\vec{y}^{(t)})$ (using parameters $p, r$ and universe $\hat{d}$)
  **For** $j \in [d]$
    $\hat{j} \leftarrow h^{(t)}(j)$
    $z_j \leftarrow \min(z_j, \hat{z}^{(t)}_{\hat{j}})$
**Return** $\vec{z}$

overestimate. Using $T$ hash functions and taking the minimum over the $T$ estimates, we can drive down the probability that the minimum is an overestimate.

**Theorem 2.2.8.** *Fix any $\varepsilon = O(1)$, $n, T \in \mathbb{N}$, and $\delta < 2e^{-4}/n$. There are choices for parameters $p, r, \hat{d}$ such that $\mathcal{P}_{\mathsf{HIST}2}$ has the following properties:*

*i. Each user sends $T \cdot n \cdot O(d^{1/T})$ messages, each consisting of $O(\log T + \log n + \frac{1}{T}\log d)$ bits.*

*ii. $\mathcal{P}_{\mathsf{HIST}2}$ is $(2T\varepsilon, 2T\delta)$-robustly shuffle private for $n$ users.*

*iii. On any input $\vec{x} \in [d]^n$, $\mathcal{P}_{\mathsf{HIST}2}$ reports $\vec{z}$ such that*

$$\|\vec{z} - c(\vec{x})\|_\infty = O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$$

*with probability $\geq 1 - n\delta - (1/100)^T$.*

*Proof.* We choose $p, r$ exactly as in Theorem 2.2.7. We assign $\hat{d} \leftarrow \lceil n \cdot (100d)^{1/T} \rceil$. Part *i* is immediate from this choice. Part *ii* follows from basic composition (Fact 1.3.3), since the randomizer executes $\mathcal{P}_{\mathsf{HIST}}$ exactly $T$ times on user data.

To prove Part *iii*, let $E_j$ denote the event that there is a hash function $h^{(t)}$ such that a user's value $j$ experiences no collisions with another user: formally, $\exists t\ \forall j' \in \vec{x}, j' \neq j\ h^{(t)}(j) \neq h^{(t)}(j')$. When this event occurs, observe that the count of $h^{(t)}(j)$ in the hashed dataset is precisely the count of $j$ in the original dataset. Otherwise, the count of $h^{(t)}(j)$ is at least as large as $j$. Given that the analyzer $\mathcal{A}_{\mathsf{HIST}}$ reports estimates with max error $\alpha = O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ with probability $1 - n\delta$, the minimum of the estimates can only be wrong by $\alpha$. Thus, it suffices to bound the probability that $E_j$ does not occur for some $j$.

$$\begin{aligned}
\mathop{\mathbb{P}}_{\vec{h}}\left[\neg E_j\right] &= \mathop{\mathbb{P}}_{\vec{h}}\left[\forall t\ \exists j' \in \vec{x}\ h^{(t)}(j) = h^{(t)}(j')\right] \\
&= \mathop{\mathbb{P}}_{\vec{h}}\left[\exists j' \in \vec{x}\ h^{(t)}(j) = h^{(t)}(j')\right]^T \\
&\leq (n \cdot \mathop{\mathbb{P}}_{\vec{h}}\left[h^{(t)}(j) = h^{(t)}(j')\right])^T \\
&= (n/\hat{d})^T = (1/100)^T \cdot \frac{1}{d} \\
\therefore \mathop{\mathbb{P}}_{\vec{h}}\left[\exists j\ \neg E_j\right] &\leq (1/100)^T
\end{aligned}$$

□

### 2.2.5 Other Histogram Protocols

There are other shuffle protocols for the histogram problem; Table 2.2 summarizes their message complexity, length per message (in bits), and bound on $\ell_\infty$ error. Most of the other protocols attempt to optimize communication complexity (total number of bits transmitted by users) which $\mathcal{P}_{\mathsf{ZSUM}}$ does not. The single-message result comes from a simple application of the amplification lemma due to Feldman McMillan and Talwar [35].

## 2.3 Support Identification and Related Problems

In this section, we consider problems that reduce to *support identification*. Our shuffle protocol $\mathcal{P}_{\mathsf{HIST}}$ allows us to solve one of these problems with arbitrarily fewer samples than any protocol the non-interactive local model. This section is adapted from joint work with Balcer [7].

We begin by defining support identification.

Table 2.2: Shuffle protocols for histograms from across the literature. $\eta$ is an arbitrary constant in the interval $(0,1)$.

| Source | Messages per user | Bits per Message | $\ell_\infty$ Error |
|---|---|---|---|
| [7] (Thm. 2.2.7) | $d+1$ | $O(\log d)$ | $O\left(\frac{1}{\varepsilon^2 n}\log\frac{1}{\delta}\right)$ |
| [38] | $O(d^\eta)$ | $O(\log d)$ | $O\left(\frac{1}{\varepsilon n}\sqrt{\log d \log\frac{1}{\delta}}\right)$ |
| | $O\left(\frac{\log^3 d}{\varepsilon^2}\log\frac{\log d}{\delta}\right)$ | $O(\log n + \log\log d)$ | $O\left(\frac{\log^{3/2} d}{\varepsilon n}\sqrt{\log\frac{\log d}{\delta}}\right)$ |
| | $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon\delta}\right)$ | $O(\log n \log d)$ | $O\left(\frac{\log d}{n}+\frac{1}{\varepsilon n}\sqrt{\log d\log\frac{1}{\varepsilon\delta}}\right)$ |
| [38] & [35] | 1 | $d$ | $O\left(\frac{\log d}{n}+\frac{1}{\sqrt{\varepsilon} n^{\frac{3}{4}}}\sqrt{\log d}\left(\log\frac{1}{\delta}\right)^{\frac{1}{4}}\right)$ |

**Definition 2.3.1.** The *support identification* problem has positive integer parameters $h \le d$. Given any data universe $\mathcal{X}$ with size $d$ and any $H \subseteq \mathcal{X}$, let $\mathbf{U}_H$ be the uniform distribution over support $H$. The set of problem instances is $\{\mathbf{U}_H : H \subseteq \mathcal{X} \text{ and } |H| = h\}$. A protocol solves the $(h,d)$*-support identification problem with sample complexity* $n$ if, given $n$ users with data independently sampled from any problem instance $\mathbf{U}_H$, it identifies $H$ with probability at least 99/100.

We now show how to solve this problem in the shuffle model.

**Claim 2.3.2.** *Fix any $\varepsilon \le 1$ and $\delta < \min(2e^{-9}, 1/200h)$. The sample complexity of the $(h,d)$-support identification problem is $O(h\log h \cdot (1/\varepsilon^2)\cdot\log(1/\delta))$ under $(\varepsilon,\delta)$-robust shuffle privacy.*

*Proof.* For the purposes of this proof, we assume there is some bijection $f$ between $\mathcal{X}$ and $[d]$ so that any reference to $j \in [d]$ corresponds directly to some $f(j) \in \mathcal{X}$ and vice versa. Consider the following protocol: execute $\mathcal{P}_{\mathsf{HIST}}$ on $n$ samples from $\mathbf{U}_H$ and then choose the items $j$ whose estimates $z_j$ are at least $\alpha+1$. We will determine the magnitude of $\alpha$ later. Privacy is immediate from closure under post-processing (Fact 1.3.5). We will prove that this new protocol returns $H$ exactly with probability at least 99/100.

Let $E_{\text{samp}}$ be the event that some element in support $H$ has frequency less than $(2\alpha+1)$ in the sample. Let $E_{\text{priv}}$ be the event that the histogram protocol estimates the frequency of some element in universe $[d]$ with error larger than $\alpha$. If neither events occur, every element in $H$ has estimated frequency at least $\alpha+1$ and every element outside $H$ has estimated frequency at most $\alpha$. Hence, it suffices to show that $E_{\text{samp}}$ and $E_{\text{priv}}$ each occur with probability $\le 1/200$.

We lower bound the probability of $E_{\text{samp}}$ via a coupon collector's argument. That is, if we have $n = O(kh\log h)$ samples from $\mathbf{U}_H$ then each element of $H$ appears at least $k$ times with probability at least 199/200. Hence we set $k = (2\alpha+1)$.

We lower bound the probability of $E_{\text{priv}}$ by simply tweaking the proof of Part (iii) of Theorem 2.2.7. Specifically, we can invoke Claim 2.2.5 to conclude that $|z_j - c_j(\vec{x})| = \alpha$ with probability $\ge 1-\delta$, for any single $j \in [d]$ and some $\alpha = O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\right)$. Although we again use $Q = \{j \in [d] : c_j(\vec{x}) > 0\}$, we now upper bound $|Q|$—the number of distinct elements—by the support size $h$ instead of sample size $n$. Thus,

$$\begin{aligned}\mathbb{P}\left[E_{\text{priv}}\right] &= \mathbb{P}\left[\exists j \in [d] \text{ s.t. } |z_j - c_j(\vec{x})| > \alpha\right] \\ &\le \sum_{j\in Q}\mathbb{P}\left[\text{ s.t. } |z_j - c_j(\vec{x})| > \alpha\right] \\ &\le h\delta \le 1/200\end{aligned}$$

So we have shown that $E_{\text{samp}}$ and $E_{\text{priv}}$ each occur with probability $\leq 1/200$. This concludes the proof. □

In the above analysis, if we had used a histogram protocol with $\ell_\infty$ error that depends on the universe size $d$, then the sample complexity would in turn depend on $d$. For example, the naive protocol based upon randomized response discussed in the end of Section 2.2.1 has error $\alpha = O((1/\varepsilon) \cdot \sqrt{\log d \cdot \log(1/\delta)})$ and therefore the sample complexity would grow with $\sqrt{\log d}$.

Having shown how to solve the support identification problem using $\mathcal{P}_{\mathsf{HIST}}$, we now describe two different problems that reduce to support identification.

### 2.3.1 Multi-party Pointer Jumping

**Definition 2.3.3** (Joseph et al. [45])**.** The *multi-party pointer jumping problem* is denoted MPJ$(s,h)$ where $s,h$ are positive integer parameters. A problem instance is $\mathbf{U}_{\{Z_1,\ldots,Z_h\}}$ where each $Z_i$ is a labeling of the nodes at level $i$ in a complete $s$-ary tree. Each label $Z_{i,j}$ is an integer in $\{0,\ldots,s-1\}$. The labeling implies a root-leaf path: if the $i$-th node in the path has label $Z_{i,j}$, then the $(i+1)$-st node in the path is the $(Z_{i,j})$-th child of the $i$-th node. A protocol *solves* MPJ$(s,h)$ *with sample complexity* $n$ if, given $n$ samples from any $\mathbf{U}_{\{Z_1,\ldots,Z_h\}}$, it identifies the root-leaf path with probability at least 99/100.

**Theorem 2.3.4.** *Fix any $\varepsilon = O(1)$ and $\delta < \min(2e^{-9}, 1/200h)$. The sample complexity of* MPJ$(s,h)$ *is $O(h \log h \cdot (1/\varepsilon^2) \cdot \log(1/\delta))$ under $(\varepsilon,\delta)$-robust shuffle privacy.*

*Proof.* As with pointer-chasing, we can solve MPJ$(s,h)$ when the support is identified. From Claim 2.3.2, this takes $O(h \log h \cdot (1/\varepsilon^2) \cdot \log(1/\delta))$ samples under shuffle privacy. □

Joseph et al. [45] give a lower bound of $\Omega(h^3/(\varepsilon^2 \log h))$ under local privacy when $s = h^4$, even allowing for sequential interactivity.[1]

### 2.3.2 Pointer Chasing

**Definition 2.3.5** (Joseph et al. [46])**.** The *pointer chasing problem* is denoted PC$(k,\ell)$ where $k,\ell$ are positive integer parameters. A problem instance is $\mathbf{U}_{\{(1,a),(2,b)\}}$ where $a,b$ are permutations of $[\ell]$. A protocol *solves* PC$(k,\ell)$ *with sample complexity* $n$ if, given $n$ independent samples from any $\mathbf{U}_{\{(1,a),(2,b)\}}$, it outputs the $k$-th integer in the sequence $a_1, b_{a_1}, a_{b_{a_1}} \ldots$ with probability at least 99/100.

**Theorem 2.3.6.** *Fix any $\varepsilon = O(1)$ and $\delta < 2e^{-9}$. The sample complexity of* PC$(k,\ell)$ *is $O((1/\varepsilon^2) \cdot \log(1/\delta))$ under $(\varepsilon,\delta)$-robust shuffle privacy.*

*Proof.* To solve PC$(k,\ell)$, note that it suffices to identify the support $\{(1,a),(2,b)\}$ and simply execute the pointer chasing ourselves. Although the universe has size $d = 2 \cdot \ell!$, the support only has size $h = 2$; from Claim 2.3.2, $\mathcal{P}_{\mathsf{HIST}}$ can identify it with just $O((1/\varepsilon^2) \cdot \log(1/\delta))$ samples, independent of $k$ and $\ell$. □

In the case where $k = 2$, Joseph et al. [46] give a lower bound of $\Omega(\ell/e^\varepsilon)$ for non-interactive local protocols. Because $\ell$ can be made arbitrarily large, there is an arbitrarily large separation between shuffle privacy and non-interactive local privacy.

[1] We remark that we do not claim a polynomial separation between the sample complexity of shuffle privacy and sequentially interactive local privacy. This would require a proof that every sequentially interactive local protocol has a counterpart in the shuffle model.

## 2.4 Distinct Elements

In this section, we take the data universe to be $\mathcal{X} = [d]$ and let $\mathsf{DE}(\vec{x})$ denote the number of distinct elements in $\vec{x} \in [d]^n$, i.e. $\mathsf{DE}(\vec{x}) := |\{j \in [d] \mid \exists i \text{ where } x_i = j\}|$. Our goal is to approximate DE, minimizing additive error:

**Definition 2.4.1** (Distinct Elements Problem)**.** An algorithm $\mathcal{M}$ solves the $(d, \alpha, \beta)$-*distinct elements problem* on input length $n$ if for all $\vec{x} \in [d]^n$, $\mathop{\mathbb{P}}_{\mathcal{M}}[|\mathcal{M}(\vec{x}) - \mathsf{DE}(\vec{x})| \le \alpha] \ge 1 - \beta$.

Notice that $\mathbb{1}[\exists i \text{ where } x_i = j]$ is precisely $\mathsf{OR}(\mathbb{1}[x_1 = j], \ldots, \mathbb{1}[x_n = j])$. As a consequence, the number of distinct elements can be decomposed as the sum

$$\mathsf{DE}(\vec{x}) = \sum_{j=1}^{d} \mathsf{OR}(\mathbb{1}[x_1 = j], \ldots, \mathbb{1}[x_n = j]) \tag{2.6}$$

Recall the template protocol $\mathcal{P}_{\mathsf{REP}}$ (Section 2.2.1): given dataset $\vec{x} \in [d]^n$, it executes another protocol on the indicator bits $\mathbb{1}[x_1 = j], \ldots, \mathbb{1}[x_n = j]$ for every $j \in [d]$. Now suppose that there exists a private protocol $\mathcal{P}_{\mathsf{OR}}$ which approximates OR. If we fill the template protocol with $\mathcal{P}_{\mathsf{OR}}$, Lemma 2.2.3 implies that we can privately compute all the OR functions in (2.6). This means we can compute an estimator with standard deviation proportional to $\sqrt{d}$.

**Theorem 2.4.2** (Informal)**.** *Fix any $\varepsilon = O(1)$, $\delta < 1$ and $n > 2$. There is an $(\varepsilon, \delta)$-robustly private shuffle protocol that solves the $(d, \alpha, 1/100)$-distinct elements problem for some*

$$\alpha = O\left(\frac{1}{\varepsilon} \cdot \min(\sqrt{d}, n^{2/3})\right)$$

Chen, Ghazi, Kumar, and Manurangsi [22] give a lower bound of $\Omega(d)$ under local privacy in the regime where $d = n$, $\varepsilon = O(1)$, and $\delta = o(1/n)$. Robust shuffle privacy therefore solves the problem with polynomially smaller error.

Most of this section will be dedicated to proving the $O(\sqrt{d}/\varepsilon)$ bound on error. Note that when $d = \omega((\varepsilon n)^2)$, the bound is trivial because the number of distinct elements is bounded by the number of users $n$. To obtain the alternate bound, we hash the universe from size $d$ to $d' = O(n^{4/3})$ and then run the protocol on the new universe $[d']$. We describe this technique in more detail in Section 2.4.3.

### 2.4.1 The OR protocol $\mathcal{P}_{\mathsf{OR}}$

In this subsection, we show how to compute OR under robust shuffle privacy. Once this is done, we will be able to plug the protocol into $\mathcal{P}_{\mathsf{REP}}$ and then approximate the number of distinct elements. Our analysis begins with the algorithm $\mathcal{M}_\varepsilon$, a basic *centrally private* solution:

$$\mathcal{M}_\varepsilon(\vec{x}) = \begin{cases} \mathbf{Ber}(1/2) & \text{if } \mathsf{OR}(\vec{x}) = 1 \\ \mathbf{Ber}(1/2e^\varepsilon) & \text{otherwise.} \end{cases} \tag{2.7}$$

**Claim 2.4.3.** *For any $\varepsilon > 0$, the algorithm $\mathcal{M}_\varepsilon$ is $\varepsilon$-differentially private.*

*Proof.* For either $b = 1$ or $b = 0$, note that

$$\frac{\mathbb{P}[\mathbf{Ber}(1/(2e^\varepsilon)) = b]}{\mathbb{P}[\mathbf{Ber}(1/2) = b]} \in \{e^{-\varepsilon}, 2 - e^{-\varepsilon}\}$$

Since $2 - e^{-\varepsilon} \in [e^{-\varepsilon}, e^\varepsilon]$, we conclude that the mechanism is $\varepsilon$-centrally private. □

The key property of the above solution is that we can simulate it in the shuffle model using a technique inspired by Balle Bell Gascón and Nissim [9]. At a high level, we show that there is some $p$ such that when each user reports a message drawn from $\mathbf{Ber}(p)$, the sum (mod 2) is distributed as $\mathbf{Ber}(1/2e^{\varepsilon})$. But note that when at least one user reports $\mathbf{Ber}(1/2)$, the sum (mod 2) is distributed as $\mathbf{Ber}(1/2)$.

Our goal is complete once we implement modular arithmetic in the shuffle model. Ishai, Kushilevitz, Ostrovsky, and Sahai [44] provide a solution: at a high level, each user generates multiple one-bit messages that are drawn uniformly at random conditioned on their sum being equal to the user's bit. When each user sends enough messages, Ishai et al. prove that an analyzer cannot distinguish between two datasets that have the same sum.

**Theorem 2.4.4** (Special Case of [44, 11, 40])**.** *There exists a shuffle protocol $\mathcal{P}_{\mathsf{MOD}} = (\mathcal{R}_{\mathsf{MOD}}, \mathcal{A}_{\mathsf{MOD}})$ that takes security parameter $\sigma > 0$ and has the following properties:*

*i. Each honest user sends $O(\sigma + \log n)$ one-bit messages.*

*ii. If all users are honest, then on any input $\vec{x} \in \{0,1\}^n$ the protocol outputs $\sum_{i=1}^{n} x_i \pmod 2$.*

*iii. Let $H$ be the set of $\gamma n$ honest users. If $\gamma n > 2$, then on any input $\vec{x} \in \{0,1\}^n$*

$$\mathrm{d}_{\mathrm{TV}}\left((\mathcal{S} \circ \mathcal{R}_{\mathsf{MOD}}^{\gamma n})(\vec{x}_H), (\mathcal{S} \circ \mathcal{R}_{\mathsf{MOD}}^{\gamma n})\left(\sum_{i \in H} x_i \pmod 2, 0, \ldots, 0\right)\right) < \gamma \cdot 2^{-\sigma}$$

Given Lemma 2.4.6 and Theorem 2.4.4, we are now ready to present our OR protocol $\mathcal{P}_{\mathsf{OR}} = (\mathcal{R}_{\mathsf{OR}}, \mathcal{A}_{\mathsf{MOD}})$. As stated previously, it takes a parameter $p$ (alongside the parameter $\sigma$). The randomizer $\mathcal{R}_{\mathsf{OR}}$ is given in Algorithm 11. The analyzer is the same one from Theorem 2.4.4.

**Algorithm 11:** Randomizer $\mathcal{R}_{\mathsf{OR}}$

**Input:** user data $x \in \{0,1\}$
**Output:** message vector $\vec{y} \in \{0,1\}^*$

**If** $x = 1$ **:** Sample $u \sim \mathbf{Ber}(1/2)$;
**Else** Sample $u \sim \mathbf{Ber}(p)$ ;
$\vec{y} \leftarrow \mathcal{R}_{\mathsf{MOD}}(u)$, with parameter $\sigma$
**Return** $\vec{y}$

**Theorem 2.4.5.** *Fix any $\varepsilon > 0$, $\delta < 1$, and $n > 4$. There exists choices of parameters $p, \sigma$ such that the shuffle protocol $\mathcal{P}_{\mathsf{OR}} = (\mathcal{R}_{\mathsf{OR}}, \mathcal{A}_{\mathsf{MOD}})$ has the following properties:*

*i. Each honest user sends $O(\log \frac{e^{\varepsilon}+1}{\delta} + \log n)$ one-bit messages.*

*ii. If all users are honest, then on any input $\vec{x} \in \{0,1\}^n$ $\mathcal{P}_{\mathsf{OR}}(\vec{x})$ has the same distribution as $\mathcal{M}_{\varepsilon}(\vec{x})$.*

*iii. If $\varepsilon > 1$, the protocol is $(\varepsilon + \ln(1/\gamma), \delta)$-robustly shuffle private. Otherwise it is $(\varepsilon^{\gamma}/\gamma, \delta)$-robustly shuffle private.*

*Proof.* We choose $\sigma \leftarrow \log \frac{e^{\varepsilon}+1}{\delta}$ and $p \leftarrow \frac{1-(1-1/e^{\varepsilon})^{1/n}}{2}$. Part *(i)* is immediate from substitution of $\sigma$ into Theorem 2.4.4. To prove the other parts, we rely on the following technical lemma:

**Lemma 2.4.6.** *Fix any integers $m \le n$ and real number $p^* \in [0, 1/2]$. If we assign $p \leftarrow \frac{1-(1-2p^*)^{1/n}}{2}$ and sample i.i.d. $X_1, \ldots, X_m \sim \mathbf{Ber}(p)$, then the sum $X := \sum_{i=1}^{m} X_i \pmod 2$ is distributed as*

$$\mathbf{Ber}\left(\frac{1-(1-2p^*)^{m/n}}{2}\right).$$

The proof is given in Appendix A.3.

Part *ii*: If $\mathsf{OR}(\vec{x}) = 1$, then there is at least one user $i$ who samples a random bit $u_i \sim \mathbf{Ber}(1/2)$. Consequently, $\sum_{i=1}^{n} u_i \bmod 2$—the output of the protocol—is distributed as $\mathbf{Ber}(1/2)$. Otherwise, Lemma 2.4.6 implies the distribution is $\mathbf{Ber}(1/2e^{\varepsilon})$. This behavior is precisely that of $\mathcal{M}_{\varepsilon}(\vec{x})$.

Part *iii*: Roughly speaking, we show that the algorithm $(\mathcal{S} \circ \mathcal{R}_{\mathsf{OR}}^{\gamma n})$ simulates $\mathcal{M}_{\varepsilon'}$ where

$$\varepsilon' := \ln\left(\frac{1}{1-(1-e^{-\varepsilon})^{\gamma}}\right).$$

$\varepsilon + \ln(1/\gamma)$ serves as an upper bound to this expression; $\varepsilon^{\gamma}/\gamma$ is an alternate bound when $\varepsilon < 1$. For the proof, see Lemma A.3.1 in Appendix A.3.

We will present a series of intermediary algorithms that incrementally change $\mathcal{M}_{\varepsilon'}$ into $(\mathcal{S} \circ \mathcal{R}_{\mathsf{OR}}^{\gamma n})$ while maintaining privacy. Throughout, we will rely on the helper algorithm $T : \{0,1\}^{\gamma n} \to \{0,1\}^{\gamma n}$: on input $\vec{x}$, this algorithm maps each $x_i$ to a random $u_i$ such that $u_i \sim \mathbf{Ber}(1/2)$ if $x_i = 1$ and $u_i \sim \mathbf{Ber}(p)$ otherwise.

We first consider the algorithm $\mathcal{M}_1(\vec{x})$, which executes $\vec{u} \leftarrow T(\vec{x})$ and then outputs $U := \sum_{i=1}^{\gamma n} u_i \pmod 2$. On every input $\vec{x}$, we claim that $\mathcal{M}_1(\vec{x}) \sim \mathcal{M}_{\varepsilon'}(\vec{x})$ and therefore $\mathcal{M}_1$ is also $\varepsilon'$-d.p. If $\mathsf{OR}(\vec{x}) = 1$, then some $x_i$ has value 1 so that $U \sim \mathbf{Ber}(1/2)$, matching $\mathcal{M}_{\varepsilon'}(\vec{x})$. Otherwise, Lemma 2.4.6 implies that

$$\begin{aligned} U &\sim \mathbf{Ber}\left(\frac{1-(1-e^{-\varepsilon})^{\gamma}}{2}\right) \\ &= \mathbf{Ber}(1/(2e^{\varepsilon'})) \end{aligned}$$

which again matches $\mathcal{M}_{\varepsilon'}(\vec{x})$.

Our second step is the algorithm $\mathcal{M}_2(\vec{x})$, which executes $U \leftarrow \mathcal{M}_1(\vec{x})$ and then runs $(\mathcal{S} \circ \mathcal{R}_{\mathsf{MOD}}^{\gamma n})$ on the input $(U, 0, \ldots, 0)$, where $\mathcal{R}_{\mathsf{MOD}}$ is the modular arithmetic randomizer. By the post-processing property of differential privacy (Fact 1.3.5), $\mathcal{M}_2(\vec{x})$ is also $\varepsilon'$-d.p.

Our third step is the algorithm $\mathcal{M}_3(\vec{x})$, which executes $\vec{u} \leftarrow T(\vec{x})$ and then runs the modular arithmetic protocol $(\mathcal{S} \circ \mathcal{R}_{\mathsf{MOD}}^{\gamma n})$ on $\vec{u}$. By Theorem 2.4.4, the following bound holds for every $\vec{x} \in \{0,1\}^{\gamma n}$:

$$\|\mathcal{M}_3(\vec{x}) - \mathcal{M}_2(\vec{x})\|_{\mathrm{TV}} \le \gamma \cdot 2^{-\sigma} = \frac{\gamma}{e^{\varepsilon}+1} \cdot \delta.$$

Given that $\mathcal{M}_3$ behaves similarly to $\mathcal{M}_2$ on every input, we invoke the following technical lemma which states that a close simulation of a pure differentially private algorithm results in an approximately differentially private algorithm:

**Lemma 2.4.7** (Lemma 1.2 [9])**.** *Let $\mathcal{M}, \mathcal{M}'$ be algorithms such that for every $\vec{w}$, $\|\mathcal{M}(\vec{w}) - \mathcal{M}'(\vec{w})\|_{\mathrm{TV}} \le \Delta$. If $\mathcal{M}$ is $(\varepsilon, 0)$-differentially private, then $\mathcal{M}'$ is $(\varepsilon, (e^{\varepsilon}+1)\cdot\Delta)$-differentially private.*

This implies that $\mathcal{M}_3$ is $(\varepsilon', \delta')$-differentially private where $\delta' = \gamma \cdot \frac{e^{\varepsilon'}+1}{e^{\varepsilon}+1} \cdot \delta$. Note that $\delta'$ is bounded above by $\delta$:

$$\begin{aligned} \delta' &= \gamma \cdot \frac{e^{\varepsilon'}+1}{e^{\varepsilon}+1} \cdot \delta \\ &\le \gamma \cdot \frac{e^{\varepsilon}/\gamma + 1}{e^{\varepsilon}+1} \cdot \delta \\ &\le \delta \end{aligned}$$

Finally, notice that $(\mathcal{S} \circ \mathcal{R}_{\mathsf{OR}}^{\gamma n})(\vec{x})$ has the same distribution as $\mathcal{M}_3(\vec{x})$. Therefore, $(\mathcal{S} \circ \mathcal{R}_{\mathsf{OR}}^{\gamma n})$ is also $(\varepsilon', \delta)$-differentially private. □

### 2.4.2 The distinct elements protocol $\mathcal{P}_{\mathsf{DE}}$

In this section, we present our distinct elements protocol. We will use the randomizer $\mathcal{R}_{\mathsf{DE}}$, which denotes the template randomizer $\mathcal{R}_{\mathsf{REP}}$ given access to $\mathcal{R}_{\mathsf{OR}}$. We will also use the analyzer $\mathcal{A}_{\mathsf{DE}}$, whose pseudocode is presented in Algorithm 12.

**Algorithm 12:** $\mathcal{A}_{\mathsf{DE}}$, analyzer for distinct elements

**Input:** $\vec{y} \in ([d] \times \mathcal{Y})^*$, where $\mathcal{Y}$ is the message space of $\mathcal{R}_{\mathsf{MOD}}$
**Output:**
$(z_1, \dots, z_d) \leftarrow \mathcal{A}_{\mathsf{REP}}(\vec{y})$, instantiated with $\mathcal{A}_{\mathsf{MOD}}$
$\hat{z} \leftarrow \sum_{j=1}^{d} z_j$
$z \leftarrow \frac{2\hat{z}e^{\varepsilon} - d}{e^{\varepsilon} - 1}$
**Return** $z$

**Theorem 2.4.8.** *Fix any $\varepsilon > 0$, $\delta < 1$, and $n > 4$. The protocol $\mathcal{P}_{\mathsf{DE}} = (\mathcal{R}_{\mathsf{DE}}, \mathcal{A}_{\mathsf{DE}})$ has the following properties:*

*i. Each user sends at most $O(d(\log \frac{e^{\varepsilon}+1}{\delta} + \log n))$ messages, each consisting of $O(\log d)$ bits.*

*ii. If $\varepsilon > 1$, the protocol is $(2\varepsilon + 2\ln(1/\gamma), 2\delta)$-robustly shuffle private. Otherwise it is $(2\varepsilon^{\gamma}/\gamma, 2\delta)$-robustly shuffle private.*

*iii. $\mathcal{P}_{\mathsf{DE}}$ solves the $(d, \alpha, \beta)$-distinct elements problem, where*

$$\alpha = \frac{e^{\varepsilon}}{e^{\varepsilon} - 1} \cdot \sqrt{2d \ln \frac{2}{\beta}}$$

*Proof.* Part *i* is immediate from Theorem 2.4.5 and the fact that $\mathcal{P}_{\mathsf{REP}}$ executes $\mathcal{P}_{\mathsf{OR}}$ $d$ times, labeling messages each time. Part *ii* is immediate from Lemma 2.2.3.

To prove Part *iii*, note that $\hat{z} = \sum_{j=1}^{k} z_j$ has expectation $\mathbb{E}[\hat{z}] = \frac{\mathsf{DE}(\vec{x})}{2} + \frac{d - \mathsf{DE}(\vec{x})}{2e^{\varepsilon}}$. In turn, the output of $\mathcal{P}_{\mathsf{DE}}$ has expectation $\mathbb{E}\left[\frac{2\hat{z}e^{\varepsilon} - d}{e^{\varepsilon} - 1}\right] = \mathsf{DE}(\vec{x})$. A Hoeffding bound implies that

$$\mathbb{P}\left[\left|\mathcal{P}_{\mathsf{DE}}(\vec{x}) - \mathsf{DE}(\vec{x})\right| > \frac{e^{\varepsilon}}{e^{\varepsilon} - 1} \cdot \sqrt{2d \ln \frac{2}{\beta}}\right] \leq \beta$$

This concludes the proof. □

### 2.4.3 Adapting the protocol to large data universes

In this section, we show how to use public randomness to obtain a protocol with error $O(n^{2/3})$. First, we derive the following bound on the error introduced by hashing.

**Lemma 2.4.9.** *Let $k, k' \in \mathbb{N}$ such that $k \geq k'$. Let $h$ be sampled uniformly from a 2-universal hash family $\mathcal{H}$ mapping $[k]$ to $[k']$. Let $S \subseteq [k]$ and $S' := \{s' \in [k'] \mid \exists s \in S \text{ s.t. } h(s) = s'\} \subseteq [k']$. Then for all $\beta \in (0, 1)$,*

$$\mathbb{P}_{h}\left[|S| - |S'| \geq \frac{|S|^2}{\beta k'}\right] \leq \beta.$$

*Proof.* Let $X := |\{(s, s') \in S^2 \mid s < s' \text{ and } h(s) = h(s')\}|$, i.e. the number of collisions when hashing the set $S$. Notice that $|S'| = |\{s' \in S \mid \forall (s \in S \text{ s.t. } s < s')\ h(s) \neq h(s')\}|$. This implies $|S| - |S'| = |\{s' \in S \mid \exists s \in S \text{ s.t. } s < s' \text{ and } h(s) = h(s')\}| \leq X$. Since $h$ is sampled uniformly from a 2-universal hash family, $\mathbb{E}[X] \leq |S|^2/k'$. The result follows by Markov's inequality. □

Given parameters $d, d' \in \mathbb{N}$ and input $x \in [d]$, let $\mathcal{R}_{\mathsf{HDE}}$ be the local randomizer that (1) consults public randomness to sample $h$ uniformly from a 2-universal hash family $\mathcal{H}$ mapping $[d]$ to $[d']$ and (2) executes $\mathcal{R}_{\mathsf{DE}}$ on $h(x)$. Note that two different users executing $\mathcal{R}_{\mathsf{HDE}}$ will obtain the same $h$ since they consult the same source of randomness. We also emphasize that hashing is strictly for utility and is not used to add privacy.

**Theorem 2.4.10.** *Fix any $\varepsilon > 0$, $\delta < 1$, and $n > 2$. Let $d' \leftarrow \lceil 2n^{4/3} \rceil$. If $d \geq d'$, then the protocol $\mathcal{P}_{\mathsf{HDE}} = (\mathcal{R}_{\mathsf{HDE}}, \mathcal{A}_{\mathsf{DE}})$ has the following properties:*

*i. Each user sends at most $O(n^{4/3}(\log \frac{e^\varepsilon+1}{\delta} + \log n))$ messages of length $O(\log n)$.*

*ii. If $\varepsilon > 1$, the protocol is $(2\varepsilon + 2\ln(1/\gamma), 2\delta)$-robustly shuffle private. Otherwise it is $(2\varepsilon^\gamma/\gamma, 2\delta)$-robustly shuffle private.*

*iii. $\mathcal{P}_{\mathsf{HDE}}$ solves the $(d, \alpha, \beta)$-distinct elements problem for*

$$\alpha = \frac{n^{2/3}}{\beta} + \frac{e^\varepsilon}{e^\varepsilon - 1} \cdot 2n^{2/3} \sqrt{\ln \frac{4}{\beta}}$$

*Proof.* The communication bound is immediate from Part *i* of Theorem 2.4.8. Next we prove the privacy guarantees. For any $h \in \mathcal{H}$ and any $\vec{x} \in [k]^n$, let $h(\vec{x}) := (h(x_1), \ldots, h(x_n))$. Let $\vec{x}, \vec{x}' \in [d]^n$ be neighboring datasets. Then $h(\vec{x})$ and $h(\vec{x}')$ are also neighboring datasets, so privacy follows from Part *ii* of Theorem 2.4.8.

We finally bound the error of the protocol. Let $\alpha_1 = \frac{n^{2/3}}{\beta}$ and $\alpha_2 = \frac{e^\varepsilon}{e^\varepsilon - 1} \cdot 2n^{2/3} \sqrt{\ln \frac{4}{\beta}}$. Lemma 2.4.9 implies $\mathbb{P}\left[\left|\mathsf{DE}(h(\vec{x})) - \mathsf{DE}(\vec{x})\right| \geq \alpha_1\right] < \beta/2$. Also, Part *iii* of Theorem 2.4.8 implies $\mathbb{P}\left[\left|\mathcal{P}_{\mathsf{DE}}(h(\vec{x})) - \mathsf{DE}(h(\vec{x}))\right| \geq \alpha_2\right] < \beta/2$. By a union bound, we have that $\mathbb{P}\left[\left|\mathcal{P}_{\mathsf{HDE}}(\vec{x}) - \mathsf{DE}(\vec{x})\right| \geq \alpha_1 + \alpha_2\right] < \beta$. □

## 2.5 Uniformity Testing

In this section, we assume i.i.d. sample access to an unknown distribution $\mathbf{D}$ over the finite universe $\mathcal{X} = [d]$; a uniformity tester uses these samples to distinguish the cases where the distribution is uniform or far from uniform. Formally,

**Definition 2.5.1** (Uniformity Testing)**.** An algorithm $\mathcal{M}$ solves *$\alpha$-uniformity testing* with sample complexity $m$ when:

- If $\vec{x} \sim \mathbf{U}^m$, then $\mathbb{P}[\mathcal{M}(\vec{x}) = \text{“uniform”}] \geq 2/3$, and
- If $\vec{x} \sim \mathbf{D}^m$ where $\|\mathbf{D} - \mathbf{U}\|_{\mathrm{TV}} > \alpha$, then $\mathbb{P}[\mathcal{M}(\vec{x}) = \text{“not uniform”}] \geq 2/3$

where the probabilities are taken over the randomness of $\mathcal{M}$ and $\vec{x}$.

Note that achieving an overall 2/3 success probability is essentially equivalent to achieving $\Omega(1)$ separation between the probabilities of outputting “uniform” given uniform and non-uniform samples. This is because any such $\Delta$ separation can be amplified using $O\left(\frac{1}{\Delta^2}\right)$ repetitions. For this reason, we generally focus on achieving any such constant separation.

When proving privacy statements, we will consider the number of data points $n$ to be deterministic and their values to be arbitrary. This standard approach decouples privacy from distributional assumptions. But when bounding sample complexity, we rely on *Poissonization*: if $n \sim \mathbf{Pois}(m)$, the count of different elements $j, j' \in [d]$ are independent over the randomness of drawing $n$ which will greatly simplify the analysis. Although Poissonization impacts the sample complexity, concentration ensures that $n$ is approximately $m$ (Lemma 2.1.3). This means we can guarantee $O(m)$ samples at the cost of a constant

decrease in success probability.[2] Because we generally focus on constant separations, we typically elide the distinction between "sample complexity $m$" and "sample complexity distributed as $\mathbf{Pois}(m)$".

This section describes the uniformity testing protocol originally presented in joint work with Balcer Joseph and Mao [8]. At a high level, our protocol imitates the pan-private uniformity tester of Amin Joseph and Mao [6] (which itself imitates a centrally private uniformity tester suggested by Cai Daskalakis and Kamath [19]). This tester maintains $d$-bin histogram, one for each element, and compares a $\chi^2$-style statistic of the counts to a threshold to determine its decision. To ensure privacy, the algorithm adds Laplace noise to each bin before computing the statistic.

Our shuffle protocol computes the same test statistic as [6] but this time using the values reported by the private histogram template $\mathcal{P}_{\mathsf{REP}}$ (Section 2.2.1). We realize the template with the binary sum protocol $\mathcal{P}_{\mathsf{SYM}}$ (Section 2.1). Recall that it adds symmetric noise for privacy like the Laplace mechanism; we leverage this symmetry to show that our uniformity tester has sample complexity $O(d^{3/4})$. We then apply a binning trick—roughly, maintaining coarser counts for random groups of elements rather than every element separately—to obtain our final uniformity tester with sample complexity $O(d^{2/3})$ (Section 2.5.2).

**Theorem 2.5.2** (Informal)**.** *Fix any $\varepsilon = O(1)$, and $0 < \alpha, \delta < 1$. There exists a protocol that is $(\varepsilon, \delta)$-robustly shuffle private and solves $\alpha$-uniformity testing with sample complexity*

$$m = O\left(\left(\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{d^{1/2}}{\alpha\varepsilon} + \frac{d^{1/2}}{\alpha^2}\right) \cdot \ln^{1/2}\left(\frac{1}{\delta}\right)\right).$$

For comparison, Acharya Canonne Freitag and Tyagi [1] uniformity testing under local privacy demands $\Omega(\frac{d}{\alpha^2\varepsilon^2})$ samples .

### 2.5.1 Preliminary Uniformity Tester

In this section, we give a preliminary private protocol for $\alpha$-uniformity testing. It first compiles private estimates of the sample counts for each element in $[d]$. This is achieved by running the SYM binary sum protocol inside the REP template; specifically, the randomizer is

$$\mathcal{R}_{\mathsf{UT}}(\cdot) := \mathcal{R}_{\mathsf{REP}}(\cdot, \mathcal{R}_{\mathsf{SYM}}).$$

The analyzer then uses the private counts to compute a statistic $Z'$; the pseudocode is presented in Algorithm 13. We will show that $Z'$ is larger than a threshold $t$ when the underlying distribution is sufficiently non-uniform and smaller when equal to uniform. The result is that the tester has sample complexity scaling with $d^{3/4}$; the next section shows how to use this tester as a black box to obtain a tester with sample complexity scaling with $d^{2/3}$.

**Algorithm 13:** $\mathcal{A}_{\mathsf{UT}}$, an analyzer for private uniformity testing

**Input:** A message vector $\vec{y} \in ([d] \times \{0,1\})^*$
**Output:** A string in {"uniform", "not uniform"}
$\vec{z} \leftarrow \mathcal{A}_{\mathsf{REP}}(\vec{y}, \mathcal{A}_{\mathsf{SYM}})$
Compute statistic $Z' \leftarrow \frac{d}{m}\sum_{j=1}^{d}((z_j - m/d)^2 - z_j)$
**Return** "not uniform" *if* $Z' > t$ *otherwise* "uniform"

[2] We briefly sketch the argument. Let $\mathcal{T}$ denote a tester with Poissonized sample complexity $m$. Now consider the algorithm $\mathcal{T}'$ that takes $100m$ samples from the distribution $\mathbf{D}$ and feeds a random subset of $n$ to $\mathcal{T}$, where $n \leftarrow \min(100m, \mathbf{Pois}(m))$. Lemma 2.1.3 implies that the statistical distance between $\mathcal{T}(\mathbf{D}^{\mathbf{Pois}(m)})$ and $\mathcal{T}'(\mathbf{D}^{100m})$ is at most a constant.

**Theorem 2.5.3.** *For any $\varepsilon = O(1)$ and $0 < \alpha, \delta < 1$, there exists parameters $\lambda > 0$ and $t \in \mathbb{R}$ such that the protocol $\mathcal{P}_{\mathsf{UT}} = (\mathcal{R}_{\mathsf{UT}}, \mathcal{A}_{\mathsf{UT}})$ is $\left(2\varepsilon/\sqrt{\gamma}, 2\delta\right)$-robustly shuffle private and solves $\alpha$-uniformity testing with sample complexity*

$$O\left(\frac{d^{3/4}}{\alpha\varepsilon}\ln^{1/2}\left(\frac{1}{\delta}\right) + \frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}\ln^{1/3}\left(\frac{1}{\delta}\right) + \frac{d^{1/2}}{\alpha^2}\right).$$

*Proof.* We will choose $\lambda$ exactly as in Claim 2.1.6 so that the robust shuffle privacy guarantee follows immediately from Lemma 2.2.3. Thus, it remains to bound the sample complexity of the protocol.

Recall that we use $c_j(\vec{x})$ to denote the count of $j$ in $\vec{x}$. Let $\eta_j = z_j - c_j(\vec{x})$, the error in the estimate of $c_j(\vec{x})$. We rewrite $Z'$ in terms of $\eta_j$:

$$\begin{aligned}
Z' &= \frac{d}{m}\sum_{j=1}^{d}\left[\left(z_j - \frac{m}{d}\right)^2 - z_j\right] \\
&= \frac{d}{m}\sum_{j=1}^{d}\left[\left(c_j(\vec{x}) + \eta_j - \frac{m}{d}\right)^2 - (c_j(\vec{x}) + \eta_j)\right] && \text{(By definition)} \\
&= \underbrace{\frac{d}{m}\sum_{j=1}^{d}\left[\left(c_j(\vec{x}) - \frac{m}{d}\right)^2 - c_j(\vec{x})\right]}_{Z} + \underbrace{\frac{d}{m}\sum_{j=1}^{d}\eta_j^2}_{A} + \underbrace{\frac{2d}{m}\sum_{j=1}^{d}\eta_j\cdot\left(c_j(\vec{x}) - \frac{m}{d}\right)}_{B} - \underbrace{\frac{d}{m}\sum_{j=1}^{d}\eta_j}_{C}
\end{aligned}$$

We will show that, when $\mathbf{D} = \mathbf{U}$, this sum is below threshold $t$ with probability $\geq 9/10$. When $\|\mathbf{D} - \mathbf{U}\|_{\mathrm{TV}} > \alpha$, it is below $\tau$ with probability $\leq 3/5$. The difference between these probabilities is a constant, which suffices for our goal of constant success probability.

Case 1: $\mathbf{D} = \mathbf{U}$. Here, we shall bound each of the four terms $Z, A, B, C$. For sake of clarity, this section will only bound the term $B$; we defer analyses of $Z, A, C$ to Appendix A.4.

We first derive the expected value of $B$:

$$\begin{aligned}
\mathbb{E}[B] &= \frac{2d}{m}\sum_{j=1}^{d}\mathbb{E}\left[\eta_j\cdot(c_j(\vec{x}) - m/d)\right] \\
&= \frac{2d}{m}\sum_{j=1}^{d}\mathbb{E}\left[\eta_j\right]\cdot\mathbb{E}\left[c_j(\vec{x}) - m/d\right] && \text{(Independence)} \\
&= 0
\end{aligned}$$

The final equality comes from the fact that $c_j(\vec{x}) \sim \mathbf{Pois}(m/d)$. Next, we obtain its variance:

$$\begin{aligned}
\mathrm{Var}[B] &= \frac{4d^2}{m^2}\sum_{j=1}^{d}\mathrm{Var}\left[\eta_j\cdot(c_j(\vec{x}) - m/d)\right] && \text{(Independence)} \\
&= \frac{4d^2}{m^2}\sum_{j=1}^{d}\mathbb{E}\left[\eta_j^2\right]\cdot\mathbb{E}\left[(c_j(\vec{x}) - m/d)^2\right] - \mathbb{E}\left[\eta_j\right]^2\cdot\mathbb{E}\left[c_j(\vec{x}) - m/d\right]^2 && \text{(Independence)} \\
&= \frac{4d}{m}\sum_{j=1}^{d}\mathbb{E}\left[\eta_j^2\right] && (\mathbf{D} = \mathbf{U}) \\
&= \frac{d^2\lambda}{m}
\end{aligned}$$

The final equality comes from Claim 2.1.7. Via Chebyshev's inequality,

$$\mathbb{P}\left[B > \sqrt{\frac{40d^2\lambda}{m}}\right] \leq 1/40. \tag{2.8}$$

In Appendix A.4, we use similar steps to arrive at the following set of bounds:

**Claim 2.5.4.** *Sample $n \sim \mathbf{Pois}(m)$ and $\vec{x} \sim \mathbf{U}^n$. There is a constant $\kappa$ such that when $m > \kappa d^{1/2}/\alpha^2$, the following inequalities hold in an execution of $\mathcal{P}_{\mathsf{UT}}(\vec{x})$:*

$$\mathbb{P}\left[Z > \frac{3\alpha^2 m}{250}\right] < 1/40$$
$$\mathbb{P}\left[A > \frac{d^2\lambda}{4m} + \sqrt{\frac{20d^3\lambda^2}{m^2}}\right] < 1/40$$
$$\mathbb{P}\left[C < -\sqrt{\frac{10d^3\lambda}{m^2}}\right] < 1/40$$

By a union bound over (2.8) and Claim 2.5.4, the following holds with probability $\geq 9/10$:

$$\begin{aligned} Z' &= Z + A + B - C \\ &< \underbrace{\frac{3\alpha^2 m}{250}}_{Z} + \underbrace{\frac{d^2\lambda}{4m} + \sqrt{\frac{20d^3\lambda^2}{m^2}}}_{A} + \underbrace{\sqrt{\frac{40d^2\lambda}{m}}}_{B} + \underbrace{\sqrt{\frac{10d^3\lambda}{m^2}}}_{C} \end{aligned} \tag{2.9}$$

We will set threshold parameter $t$ to the right-hand side of the above inequality. Thus, given uniform samples, the protocol has probability at least $9/10$ of correctly answering "uniform."

Case 2: $\|\mathbf{D} - \mathbf{U}\|_{\mathrm{TV}} > \alpha$. Here, we lower bound the four terms $Z, A, B, C$. Again, we focus on the term $B$; the analyses of $Z, A, C$ are deferred to Appendix A.4.

Using the same steps as in Case 1, we find that the expectation of $B$ is again 0. However, the variance becomes a difficult quantity to bound because it is a function of $\mathbb{E}\left[(c_j(\vec{x}) - m/k)^2\right]$. This quantity depends on the identity of distribution $\mathbf{D}$ but all we have is the fact that $\|\mathbf{D} - \mathbf{U}\|_{\mathrm{TV}} > \alpha$. Thus, we shall use a different technique to lower bound $B$.

From Claim 2.1.7, the distribution of each $\eta_j$ is symmetric with mean zero. So for any input dataset $\vec{x}$, $B$ is a linear combination of random variables $\eta_j$ that are symmetrically distributed about zero. In Appendix A.4, we prove the following:

**Claim 2.5.5.** *Let $\eta_1, \ldots, \eta_d$ be independent random variables where each $\eta_j$ is symmetrically distributed over the set $\{\ldots, -3/2, -1, -1/2, 0, 1/2, 1, 3/2, \ldots\}$ with mean zero. For any coefficients $a_1, \ldots, a_d \in \mathbb{R}$, the random variable $\sum_{j=1}^{d} \eta_j \cdot a_j$ is symmetrically distributed with mean zero.*

Hence, we have

$$\mathbb{P}[B \geq 0] \geq 1/2. \tag{2.10}$$

To bound $Z$, $A$, and $C$, we follow the Chebyshev-based recipe:

**Claim 2.5.6.** *Sample $n \sim \mathbf{Pois}(m)$ and $\vec{x} \sim \mathbf{D}^n$ where $\|\mathbf{D} - \mathbf{U}\|_{\mathrm{TV}} > \alpha$. There is a constant $c$ such that when $m > cd^{1/2}/\alpha^2$, the following inequalities hold in an execution of $\mathcal{P}_{\mathsf{UT}}(\vec{x})$:*

$$\mathbb{P}\left[Z < \frac{\alpha^2 m}{15}\right] < 1/30$$

$$\mathbb{P}\left[A < \frac{d^2\lambda}{4m} - \sqrt{\frac{15d^3\lambda^2}{m^2}}\right] < 1/30$$

$$\mathbb{P}\left[C > \sqrt{\frac{15d^3\lambda}{2m^2}}\right] < 1/30$$

By a union bound over (2.10) and Claim 2.5.6, the following is true with failure probability $\leq 3/5$:

$$Z' > \underbrace{\frac{\alpha^2 m}{15}}_{Z} + \underbrace{\frac{d^2\lambda}{4m} - \sqrt{\frac{15d^3\lambda^2}{m^2}}}_{A} + \underbrace{0}_{B} - \underbrace{\sqrt{\frac{15d^3\lambda}{2m^2}}}_{C} \tag{2.11}$$

We will prove that the right-hand side of (2.11) is larger than $t$. Since we only output "uniform" when $Z' < \tau$, this means the probability of erroneously reporting "uniform" is $\leq 3/5$.

$$\begin{aligned}
\mathrm{RHS}(2.11) - t &= \frac{\alpha^2 m}{15} + \frac{d^2\lambda}{4m} - \sqrt{\frac{15d^3\lambda^2}{m^2}} - \sqrt{\frac{15d^3\lambda}{2m^2}} \\
&\quad - \frac{3\alpha^2 m}{250} - \frac{d^2\lambda}{4m} - \sqrt{\frac{20d^3\lambda^2}{m^2}} - \sqrt{\frac{40d^2\lambda}{m}} - \sqrt{\frac{10d^3\lambda}{m^2}} \\
&> \frac{4\alpha^2 m}{75} - \frac{9d^{3/2}\lambda}{m} - \frac{7d\lambda^{1/2}}{m^{1/2}} - \frac{6d^{3/2}\lambda^{1/2}}{m} \\
&= \underbrace{\frac{2\alpha^2 m}{75} - \frac{9d^{3/2}\lambda}{m} - \frac{6d^{3/2}\lambda^{1/2}}{m}}_{V} + \underbrace{\frac{2\alpha^2 m}{75} - \frac{7d\lambda^{1/2}}{m^{1/2}}}_{W}
\end{aligned}$$

It will suffice to show that both $V, W$ are larger than 0. For some constant $\kappa_1$, if $m > \frac{\kappa_1}{\alpha} \cdot d^{3/4} \cdot \lambda^{1/2}$ then $V > 0$. For some constant $\kappa_2$, if $m > \frac{\kappa_2}{\alpha^{4/3}} d^{2/3}\lambda^{1/3}$ then $W > 0$. □

### 2.5.2 Final Uniformity Tester

We now use a technique due to Acharya, Canonne, Han, Sun, and Tyagi [2] and Amin et al. [6] (itself a generalization of a similar technique from Acharya et al. [1]) to reduce the sample complexity dependence on $d$ from $d^{3/4}$ to $d^{2/3}$. The idea is to reduce the size of the data universe $[d]$ by grouping random elements and then performing the test on the smaller universe $[\hat{d}]$. The randomized grouping also reduces testing distance—partitions may group together elements with non-uniform mass to produce a group with near-uniform overall mass, thus hiding some of the original distance—but the reduction in universe size outweighs this side-effect.

We first introduce some notation. Given a partition $G$ of $[d]$ into $G_1, \ldots, G_{\hat{d}}$, let $\mathbf{D}_G$ denote the distribution over $[\hat{d}]$ such that the probability of sampling $\hat{j}$ from $\mathbf{D}_G$ is equal to the probability that $j \in G_{\hat{j}}$ where $j \sim \mathbf{D}$. Formally, $\mathbb{P}\left[\mathbf{D}_G = \hat{j}\right] = \sum_{j \in G_{\hat{j}}} \mathbb{P}[\mathbf{D} = j]$.

Our work will rely on the following lemma:

**Lemma 2.5.7** (Domain Compression [2, 6])**.** *Let $\mathbf{D}$ be a distribution over $[d]$ such that $\|\mathbf{D} - \mathbf{U}\|_{\mathrm{TV}} = \alpha$. If $G$ is a uniformly random of $[d]$ into $\hat{d}$ groups, then with probability $\geq 1/954$ over $G$,*

$$\|\mathbf{D}_G - \mathbf{U}\|_{\mathrm{TV}} \geq \alpha \cdot \frac{\sqrt{\hat{d}}}{477\sqrt{10d}}.$$

Applying this trick to reduce $[d]$ to $[\hat{d}]$ and then running our initial protocol $\mathcal{P}_{\mathsf{UT}} = (\mathcal{R}_{\mathsf{UT}}, \mathcal{A}_{\mathsf{UT}})$ on $[\hat{d}]$ with distance parameter $\hat{\alpha} = \alpha \frac{\sqrt{\hat{d}}}{477\sqrt{10d}}$ gives our final uniformity tester. The given asymptotic bound requires different values of $\hat{d}$ depending on parameter settings; these appear in the proof.

**Theorem 2.5.8.** *Fix any $\varepsilon = O(1)$, and $0 < \alpha, \delta < 1$. There exists a protocol that is $\left(2\varepsilon/\sqrt{\gamma}, 2\delta, 1/2\right)$-robustly shuffle private and solves $\alpha$-uniformity testing with sample complexity*

$$m = O\left(\left(\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{d^{1/2}}{\alpha\varepsilon} + \frac{d^{1/2}}{\alpha^2}\right) \cdot \ln^{1/2}\left(\frac{1}{\delta}\right)\right).$$

*Proof.* The protocol assigns $\hat{d}$ according to the following rule:

$$\hat{d} = \begin{cases} 2 & \text{if } \frac{d^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} < 2 \\ d & \text{if } \frac{d^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} > d \\ \frac{d^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} & \text{otherwise} \end{cases}$$

The users and analyzer determine the partition $G$ using public randomness. User $i$ reports $\mathcal{R}_{\mathsf{UT}}(w_i)$ where $w_i = \hat{j}$ iff $x_i \in G_{\hat{j}}$. The analyzer executes $\mathcal{A}_{\mathsf{UT}}$ but using the new dimension $\hat{d}$ and new error parameter $\hat{\alpha}$. Privacy is immediate from Theorem 2.5.3, so it remains to derive the sample complexity.

Suppose we could prove the following two statements for some large enough $m$:

- If users generate samples $\vec{w}$ from $\mathbf{U}$, then $\mathbb{P}[\mathcal{P}_{\mathsf{UT}}(\vec{w}) = \text{“uniform”}] \geq 1 - 10^{-4}$, and
- If users generate samples $\vec{w}$ from $\mathbf{D}_G$ where $\|\mathbf{D}_G - \mathbf{U}\|_{\mathrm{TV}} > \hat{\alpha}$, then $\mathbb{P}[\mathcal{P}_{\mathsf{UT}}(\vec{w}) = \text{“not uniform”}] \geq 1 - 10^{-4}$

If $\|\mathbf{D} - \mathbf{U}\|_{\mathrm{TV}} > \alpha$, then by Lemma 2.5.7, $\|\mathbf{D}_G - \mathbf{U}\|_{\mathrm{TV}} < \hat{\alpha}$ with probability $< 953/954$. In the event that $\|\mathbf{D}_G - \mathbf{U}\|_{\mathrm{TV}} \geq \hat{\alpha}$, the second bullet above tells us that the tester returns “uniform” with probability $< 10^{-4}$. So by a union bound, the the tester returns “uniform” with probability $< 10^{-4} + 953/954$.

But if $\mathbf{D} = \mathbf{U}$, then $\mathbf{D}_G = \mathbf{U}$ and so the probability of “uniform” is $\geq 1 - 10^{-4}$. Because $1 - 10^{-4} - (10^{-4} + 953/954)$ is a positive constant, we can distinguish the two cases with large enough $m$.

So it remains to prove the two bullet points for large $m$. They will naturally follow from the sample complexity guarantee of $\mathcal{P}_{\mathsf{UT}}$ (Theorem 2.5.3) for dimension $\hat{d}$ and error $\hat{\alpha}$:

$$\begin{aligned} m &= O\left(\frac{\hat{d}^{3/4}}{\hat{\alpha}\varepsilon}\ln^{1/2}\left(\frac{1}{\delta}\right) + \frac{\hat{d}^{2/3}}{\hat{\alpha}^{4/3}\varepsilon^{2/3}}\ln^{1/3}\left(\frac{1}{\delta}\right) + \frac{\hat{d}^{1/2}}{\hat{\alpha}^2}\right) \\ &= O\left(\left(\underbrace{\frac{d^{1/2}\hat{d}^{1/4}}{\alpha\varepsilon}}_{T_1} + \underbrace{\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}}_{T_2} + \underbrace{\frac{d}{\alpha^2\hat{d}^{1/2}}}_{T_3}\right) \cdot \ln^{1/2}\left(\frac{1}{\delta}\right)\right) \qquad \text{(Value of } \hat{\alpha}\text{)} \end{aligned}$$

We split into cases based on $\hat{d}$.

Case 1: $\hat{d} = 2$. Then $\frac{d^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} < 2$, so $d^{1/2} = O\left(\frac{\alpha}{\varepsilon}\right)$ and $d^{1/6} = O\left(\frac{\alpha^{1/3}}{\varepsilon^{1/3}}\right)$. Thus,

$$\begin{aligned} T_1 + T_2 + T_3 &= O\left(\frac{d^{1/2}}{\alpha\varepsilon} + \frac{d^{1/2}\cdot d^{1/6}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{d^{1/2}d^{1/2}}{\alpha^2}\right) \\ &= O\left(\frac{d^{1/2}}{\alpha\varepsilon} + \frac{d^{1/2}}{\alpha\varepsilon} + \frac{d^{1/2}}{\alpha\varepsilon}\right) \\ &= O\left(\frac{d^{1/2}}{\alpha\varepsilon}\right). \end{aligned}$$

Case 2: $\hat{d} = d$. This means $d < \frac{d^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}}$, so $d^{3/4} < \frac{d^{1/2}\varepsilon}{\alpha}$ and $d^{1/6} < \frac{\varepsilon^{2/3}}{\alpha^{2/3}}$. Thus,

$$\begin{aligned} T_1 + T_2 + T_3 &= O\left(\frac{d^{3/4}}{\alpha\varepsilon} + \frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{d^{1/2}}{\alpha^2}\right) \\ &= O\left(\frac{d^{1/2}}{\alpha^2} + \frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}\right) \\ &= O\left(\frac{d^{1/2}}{\alpha^2} + \frac{d^{1/2}\cdot d^{1/6}}{\alpha^{4/3}\varepsilon^{2/3}}\right) \\ &= O\left(\frac{d^{1/2}}{\alpha^2}\right). \end{aligned}$$

Case 3: $\hat{d} = \frac{d^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}}$. By substitution,

$$\begin{aligned} T_1 + T_2 + T_3 &= O\left(\frac{d^{1/2}(d^{2/3}\varepsilon^{4/3}\alpha^{-4/3})^{1/4}}{\alpha\varepsilon} + \frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{d}{\alpha^2(d^{2/3}\varepsilon^{4/3}\alpha^{-4/3})^{1/2}}\right) \\ &= O\left(\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}\right) \end{aligned}$$

The claimed bound follows by taking the sum over the three cases. □

# Chapter 3

# The Limits of Robust Shuffle Privacy

The previous chapter gave robustly shuffle private protocols to solve a catalog of problems. These protocols imply separations between local privacy and robust shuffle privacy. In this chapter, we prove lower bounds that imply separations between central privacy and robust shuffle privacy. This will cement robust shuffle privacy as having an intermediate degree of strength compared to the two established notions.

Our lower bounds will use the following proof structure: transform a robustly private shuffle protocol to an internally private online algorithm and then invoke lower bounds for internal privacy. The transformation we present is a simplification of the transformation from robust shuffle privacy to pan-privacy found in joint work with Balcer et al. [8] and Ullman [24].

**Notational Conventions** We reserve capital letters in plain math text to denote random variables. For example, $S_i$ will denote the internal state of an online algorithm after processing user $i$'s data.

## 3.1 Distinct Elements

In this section, we will prove the following lower bound for the distinct elements problem (Defn. 2.4.1) under robust shuffle privacy.

**Theorem 3.1.1** (Informal)**.** *Fix any $n \geq 2d$, $\varepsilon = O(1)$, $\delta = o(\varepsilon/n)$ and a constant $\beta$. Suppose there exists a shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ that solves the $(d, \alpha, \beta)$-distinct elements problem and is $(\varepsilon, \delta)$-robustly private for $n$ users. Then $\alpha = \Omega\left(\sqrt{d/\varepsilon} + 1/\varepsilon\right)$.*

Our proof is the combination of a transformation from robust shuffle privacy to internal privacy and a lower bound for internally private distinct elements. We state the transformation first:

**Lemma 3.1.2.** *Suppose there exists a shuffle protocol $\mathcal{P}$ that solves the $(d, \alpha, \beta)$-distinct elements problem and is $(\tilde{\varepsilon}, \tilde{\delta})$-robustly private for $n$ users. Then there exists an online algorithm $\mathcal{Q}_{\mathcal{P}}$ (Algorithm 14) that solves the $(d, \alpha, \beta)$-distinct elements problem and is $(\tilde{\varepsilon}(1/2), \tilde{\delta}(1/2))$-internally private for $n/2$ users.*

At a high level, the online algorithm[1] uses the shuffle protocol to maintain a set of shuffle protocol messages as its internal state. More concretely, the online algorithm initializes its internal state using $n/2$ draws from the protocol randomizer $\mathcal{R}(1)$, processes the stream $\vec{x}$ by adding $\mathcal{R}(x_1), \ldots, \mathcal{R}(x_{n/2})$ to its collection of messages, and finally applies the protocol analyzer $\mathcal{A}$ to this final internal state to produce

[1] Although it does not strictly follow the syntax in Definition 1.3.17, it is not hard to see that $\mathcal{Q}_{\mathcal{P}}$ is a compact version of an online algorithm: the internal update algorithm is contained within the for-loop and the output algorithm consists of the final line.

output. Internal privacy follows from the original protocol's robust shuffle privacy combined with our incorporation of "dummy" messages into the state. By the original protocol's accuracy guarantee, these dummy messages – all generated from a single element– increase final error by at most 1.

We remark that the construction assumes $n$ is even, but this constraint can be removed by using $\lceil n/2 \rceil$ and $\lfloor n/2 \rfloor$ where appropriate. We avoid this technicality for sake of clarity.

**Algorithm 14:** $\mathcal{Q}_{\mathcal{P}}$, an online algorithm for distinct elements

**Input:** Data stream $\vec{x} \in [d]^{n/2}$; a shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ for distinct elements
**Output:** An integer in $[d]$
Initialize internal state $S_0$ as the messages produced by $(\mathcal{S} \circ \mathcal{R}^{n/2})(1, \ldots, 1)$
**For** $i \in [n/2]$
  Let $S_i$ be the (shuffled) union of $S_{i-1}$ and $\mathcal{R}(x_i)$
**Return** $\mathcal{A}(S_{n/2})$

*Proof of Lemma 3.1.2.* Privacy: The main idea of the proof is that, by the robust shuffle privacy of $\mathcal{P}$, the first draw from $(\mathcal{S} \circ \mathcal{R}^{n/2})(\vec{1})$ ensures privacy of any internal state view. We make this explicit below.

Consider a neighboring pair $\vec{x} \sim \vec{x}'$. We will use $S_t$ (resp. $S'_t$) to denote the internal state at time $t$ as generated by $\mathcal{Q}_{\mathcal{P}}(\vec{x})$ (resp. $\mathcal{Q}_{\mathcal{P}}(\vec{x}')$). Observe that $S_t \sim (\mathcal{S} \circ \mathcal{R}^{n/2+t})(1, \ldots, 1, x_1, \ldots, x_t)$ and $S_t \sim (\mathcal{S} \circ \mathcal{R}^{n/2+t})(1, \ldots, 1, x'_1, \ldots, x'_t)$.

Let $i$ be the index on which $\vec{x}$ and $\vec{x}'$ differ. If $i > t$, then $S_t$ must have the same distribution as $S'_t$. Otherwise, we have two executions of $(\mathcal{S} \circ \mathcal{R}^{n/2+t})$ where the inputs differ on one index. Because the number of executions of $\mathcal{R}$ is at least $n/2$, the robust shuffle privacy of $\mathcal{P}$ implies $(\tilde{\varepsilon}(1/2), \tilde{\delta}(1/2))$ -internal privacy.

Accuracy: Consider the vector $\vec{w} = (1, \ldots, 1, x_1, \ldots, x_{n/2}) \in [d]^n$. By the accuracy guarantee of the original shuffle protocol $\mathcal{P}$, we have $\mathbb{P}[|\mathcal{P}(\vec{w}) - \mathsf{DE}(\vec{w})| > \alpha] < \beta$. By the construction of $\mathcal{Q}_{\mathcal{P}}$, $\mathcal{Q}_{\mathcal{P}}(\vec{x})$ is identically distributed to $\mathcal{P}(\vec{w})$. Combining the triangle inequality and $|\mathsf{DE}(\vec{x}) - \mathsf{DE}(\vec{w})| \leq 1$, we conclude $\mathcal{Q}_{\mathcal{P}}$ solves the $(\alpha + 1, \beta)$-distinct elements problem. □

Now we give a lower bound for approximate internal privacy. It is derived from Theorem 12 in [51]. Although the authors stated their result for user-level pan-privacy, the same argument works for record-level internal privacy (because they only invoke privacy of the internal state). The result is also stated for pure differential privacy, but the proof concludes by recovering, for $\omega(1)$ elements, whether or not those elements appeared in the stream; this reconstruction-style argument implies a lower bound for approximate differential privacy:

**Lemma 3.1.3** (Implicit in [51])**.** *Fix any $n \geq d$, $\varepsilon = O(1)$, $\delta = O(1/n)$ and $\delta < \beta < 0.01$. If online algorithm $\mathcal{Q}$ solves the $(d, \alpha, \beta)$-distinct elements problem and is $(\varepsilon, \delta)$-internally private for $n$ users, then $\alpha = \Omega(\sqrt{d})$.*

We can strengthen Lemma 3.1.3 to incorporate $\varepsilon$.

**Lemma 3.1.4.** *Fix any $n \geq d$, $\varepsilon = O(1)$, $\delta \ll \frac{e^\varepsilon - 1}{e^\varepsilon + 1} \cdot \exp\left(-\frac{2\varepsilon}{e^\varepsilon - 1}\right) \cdot \frac{1}{n}$ and $\frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \exp\left(\frac{2\varepsilon}{e^\varepsilon - 1}\right) \cdot \delta < \beta < 0.01$. If online algorithm $\mathcal{Q}$ solves the $(d, \alpha, \beta)$-distinct elements problem and is $(\varepsilon, \delta)$-internally private for $n$ users, then $\alpha = \Omega\left(\sqrt{d/\varepsilon} + 1/\varepsilon\right)$.*

*Proof.* The term $\Omega(1/\varepsilon)$ in our lower bound comes from central differential privacy. We devote the rest of the proof to the $\Omega(\sqrt{d/\varepsilon})$ term. We focus on the case where $\frac{e^\varepsilon - 1}{e^\varepsilon + 1} d$, $\frac{e^\varepsilon - 1}{e^\varepsilon + 1} n$, and $\frac{e^\varepsilon + 1}{e^\varepsilon - 1}$ are integers.

Let $d' = \frac{e^\varepsilon - 1}{e^\varepsilon + 1} \cdot d$, $n' = \frac{e^\varepsilon - 1}{e^\varepsilon + 1} \cdot n$, $\delta' = \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \exp\left(\frac{2\varepsilon}{e^\varepsilon - 1}\right) \cdot \delta$, and $\alpha' = \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \alpha$. Let $\mathcal{Q}'$ be the online algorithm that initializes $\mathcal{Q}$ and then, for each received element $j \in [d']$, updates $\mathcal{Q}$ with the $\frac{e^\varepsilon + 1}{e^\varepsilon - 1}$ elements $(j, 1), (j, 2), (j, 3) \ldots$ each of which can be encoded into $[d]$.

We claim $\mathcal{Q}'$ solves the $(d', \alpha', \beta)$-distinct elements problem and is $(O(1), \delta')$-internally private for $n'$ users; our lower bound comes from invoking Lemma 3.1.3.

Privacy: On neighboring streams given to $\mathcal{Q}'$, the corresponding streams given to $\mathcal{Q}$ will differ on exactly $\frac{e^\varepsilon+1}{e^\varepsilon-1}$ elements. By group privacy (Fact 1.3.2), $\mathcal{Q}'$ is $(\frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \varepsilon = O(1), \delta')$-internally private.

Accuracy: Because the stream given to it by $\mathcal{Q}'$ has length $n$, $\mathcal{Q}$ outputs an $\alpha$-accurate estimate of the number of distinct elements in that stream (with probability at least $1-\beta$). By our transformation, $\mathcal{Q}'$ can multiply the estimate by $\frac{e^\varepsilon-1}{e^\varepsilon+1}$ to get an $\alpha'$-accurate estimate of the number of distinct elements in the original input stream.

We now apply Lemma 3.1.3 to $\mathcal{Q}'$. This is possible because $\delta \ll \frac{e^\varepsilon-1}{e^\varepsilon+1} \cdot \exp\left(-\frac{2\varepsilon}{e^\varepsilon-1}\right) \cdot \frac{1}{n}$ implies $\delta' \ll 1/n$, $n \geq d$ implies $n' \geq d'$, and $\delta' < \beta < 0.01$. Thus, $O(\alpha') = \Omega(\sqrt{d'})$ so that $\alpha = \Omega(\sqrt{d/\varepsilon})$. □

We can combine Lemma 3.1.2 and Lemma 3.1.4 to obtain Theorem 3.1.1.

## 3.2 Uniformity Testing

In this section, we obtain a lower bound for uniformity testing (Defn. 2.5.1) under robust shuffle privacy.

**Theorem 3.2.1.** *For any $\varepsilon \leq 1$ and $\alpha < 1/8$, any $(\varepsilon, 0)$-robustly shuffle private protocol $\alpha$-uniformity tester has sample complexity*

$$\Omega\left(\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{\sqrt{d}}{\alpha^2} + \frac{1}{\alpha\varepsilon}\right).$$

Note that this lower bound is not directly comparable to the upper bound of Theorem 2.5.8, since the former only applies to pure differential privacy while the latter satisfies approximate differential privacy.

We derive the lower bound by using essentially the same technique for distinct elements: we create an online algorithm[2] that initializes state using dummy data and then handle new stream elements as shuffle protocol users contributing to a growing pool of repeatedly shuffled messages. Here, the dummy data consists of samples from a uniform distribution. This has the effect of diluting the true samples and worsens the testing accuracy, but to a controlled extent.

We note that the transformation can be applied to essentially any problem involving data drawn i.i.d. from a distribution (like feature selection). For this reason, we state it as a standalone lemma. For any distribution $\mathbf{D}$ over $\mathcal{X}$ and any $b \in [0,1]$, we will use $\mathbf{D}_{(b)}$ to denote the mixture $b \cdot \mathbf{D} + (1-b) \cdot \mathbf{U}$.

**Lemma 3.2.2** (Generalization of [8])**.** *For any $n > 12\ln 6$ and any $(\tilde{\varepsilon}, \tilde{\delta}, \tau)$-robustly shuffle private protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$, there exists an $(\tilde{\varepsilon}(1/2), \tilde{\delta}(1/2))$-internally private algorithm $\mathcal{Q}_\mathcal{P}$ such that*

$$\mathrm{d_{TV}}(\mathcal{Q}_\mathcal{P}(\mathbf{U}^{n/2}), \mathcal{P}(\mathbf{U}^n)) = 0 \tag{3.1}$$

*and, for any distribution $\mathbf{D}$ over $\mathcal{X}$,*

$$\mathrm{d_{TV}}(\mathcal{Q}_\mathcal{P}(\mathbf{D}^{n/2}), \mathcal{P}(\mathbf{D}^n_{(1/4)})) < 1/6. \tag{3.2}$$

*Proof.* The privacy argument is essentially identical to that of Lemma 3.1.2. The only difference is that the dummy data now consists of random values. However, since this dummy data is still independent of the true data, we can apply the same argument to translate robust shuffle privacy into internal privacy.

It remains to prove (3.1) and (3.2). In the case where user data $\vec{X}$ is drawn i.i.d. from $\mathbf{U}$, observe that every execution of $\mathcal{R}$ made by $\mathcal{Q}_\mathcal{P}$ is on an independent sample from $\mathbf{U}$. Because there are $n$ such

[2] As with Algorithm 14, Algorithm 15 does not strictly obey the syntax of Definition 1.3.17 but is a compact version of an online algorithm.

**Algorithm 15:** $\mathcal{Q}_{\mathcal{P}}$, an online algorithm built from a shuffle protocol

**Input:** Data stream $\vec{x} \in \mathcal{X}^{n/2}$; a shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ for $n$ users
Create initial state $S_0 \leftarrow (\mathcal{S} \circ \mathcal{R}^{n/2})(\mathbf{U}^{n/2})$
Sample $M \sim \mathbf{Bin}(n, 1/4)$
Set $M \leftarrow \min(M, n/2)$
**For** $i \in [n/2]$
  **If** $i \le M$ **:** $W_i \leftarrow x_i$;
  **Else** $W_i \sim \mathbf{U}$;
  Create the state $S_i$ by shuffling the messages from $S_{i-1}$ with those from $\mathcal{R}(W_i)$
**Return** $\mathcal{A}(S_{n/2})$

executions and the output of $\mathcal{Q}_{\mathcal{P}}$ is obtained by running $\mathcal{A}$ on the set of generated messages, we have that $\mathcal{Q}_{\mathcal{P}}(\mathbf{U}^{n/2})$ is equivalent in distribution to $(\mathcal{A} \circ \mathcal{S} \circ \mathcal{R}^n)(\mathbf{U}^n) = \mathcal{P}(\mathbf{U}^n)$.

Otherwise, consider $n$ samples from $\mathbf{D}_{(1/4)}$. The number of samples drawn from $\mathbf{D}$ is distributed as $\mathbf{Bin}(n, 1/4)$. By a multiplicative Chernoff bound, we have that $\mathbb{P}[\mathbf{Bin}(n, 1/4) > n/2] < 1/6$. Thus the TV distance between $\mathbf{Bin}(n, 1/4)$ and the distribution of the truncated random variable $M$ is at most $1/6$. In turn, the TV distance between

$$\mathcal{P}(\mathbf{D}^n_{(1/4)}) = \mathcal{A}(\mathcal{S}(\overbrace{\underbrace{\mathcal{R}(\mathbf{D}), \ldots, \mathcal{R}(\mathbf{D})}_{\mathbf{Bin}(n,1/4) \text{ terms}}, \mathcal{R}(\mathbf{U}), \ldots, \mathcal{R}(\mathbf{U})}^{n \text{ terms}}))$$

and

$$\mathcal{Q}_{\mathcal{P}}(\mathbf{D}^{n/2}) = \mathcal{A}(\mathcal{S}(\overbrace{\underbrace{\mathcal{R}(\mathbf{D}), \ldots, \mathcal{R}(\mathbf{D})}_{M \text{ terms}}, \mathcal{R}(\mathbf{U}), \ldots, \mathcal{R}(\mathbf{U})}^{n \text{ terms}}))$$

is at most $1/6$ as well. □

We apply Lemma 3.2.2 to the special case of uniformity testing:

**Corollary 3.2.3.** *If $\mathcal{P}$ is an $(\tilde{\varepsilon}, \tilde{\delta}, \tau)$-robustly shuffle private protocol that solves $\alpha$-uniformity testing with sample complexity $n > 12 \ln 6$, then $\mathcal{Q}_{\mathcal{P}}$ is an $(\tilde{\varepsilon}(\tau), \tilde{\delta}(\tau))$-internally private algorithm that solves $4\alpha$-uniformity testing with sample complexity $n/2$.*

*Proof.* Privacy follows immediately from Lemma 3.2.2, so we devote the rest of the proof to the accuracy claim.

For any $b$, observe that $\mathrm{d}_{\mathrm{TV}}(\mathbf{D}_{(b)}, \mathbf{U}) = b \cdot \mathrm{d}_{\mathrm{TV}}(\mathbf{D}, \mathbf{U})$. This implies that $\mathcal{Q}_{\mathcal{P}}$ solves $4\alpha$-uniformity testing whenever $\mathcal{P}$ solves $\alpha$-uniformity testing:

- On inputs drawn from $\mathbf{U}$, $\mathcal{Q}_{\mathcal{P}}$ simulates $\mathcal{P}(\mathbf{U}^n)$ exactly. Thus, it will report "uniform" with probability at least $2/3$.
- On inputs drawn from $\mathbf{D}$ where $\mathrm{d}_{\mathrm{TV}}(\mathbf{D}, \mathbf{U}) > 4\alpha$, $\mathcal{Q}_{\mathcal{P}}$ simulates $\mathcal{P}(\mathbf{D}^n_{1/4})$ to within $1/6$ in TV-distance. Thus, it will report "uniform" with probability at most $1/3 + 1/6 = 1/2$.

The distance between $2/3$ and $1/2$ is a constant which suffices for testing. □

Next, we present a lower bound for internal privacy. As with the distinct elements lower bound (Lemma 3.1.4), it was originally stated as a lower bound for pan-privacy but the argument only relies on privacy of the internal state.

**Lemma 3.2.4** (Theorem 3 from Amin et al. [6]). *For $\varepsilon = O(1)$ and $\alpha < 1/2$, any $\varepsilon$-internally private $\alpha$-uniformity tester has sample complexity*

$$\Omega\left(\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{\sqrt{d}}{\alpha^2} + \frac{1}{\alpha\varepsilon}\right).$$

Theorem 3.2.1 is immediate from Corollary 3.2.3 and Lemma 3.2.4.

## 3.3 Feature Selection and Related Problems

In this section we prove a lower bound for the feature selection problem (Defn. 2.1.13).

**Theorem 3.3.1.** *If $\mathcal{P}$ is an $(\varepsilon, \delta)$-robustly shuffle private protocol that solves $(\alpha, d)$-selection and $\delta \log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.*

The technique we use is in fact general enough to imply lower bounds for a host of problems including learning, hypothesis testing, and sparse mean estimation (see Table 1.2 and Section 3.3.5).

### 3.3.1 Technique overview

As with with the uniformity testing problem, we will leverage Lemma 3.2.2 so that it will suffice to prove lower bounds for internal privacy; we give a high-level outline of our approach to obtain these lower bounds. We will ignore the parameter $\delta$ in this discussion for brevity, but our results also apply when $\delta$ is moderately small.

For any probability distribution $\mathbf{D}$, let $\mathbf{D}^n$ be the product distribution consisting of $n$ copies of $\mathbf{D}$; in other words, a sample from $\mathbf{D}^n$ consists of $n$ i.i.d. samples from $\mathbf{D}$. Let $\{\mathbf{D}_v\}_{v\in\mathcal{V}}$ be some family of distributions over the same domain $\mathcal{X}$. Using $V$ to denote a random variable distributed uniformly over $\mathcal{V}$, $\mathbf{D}_V^n$ is the uniform mixture of product distributions.

Using $\mathbf{U}$ to denote the uniform mixture of $\{\mathbf{D}_v\}_{v\in\mathcal{V}}$, $\mathbf{U}^n$ is the product distribution consisting of $n$ copies of the uniform mixture $\mathbf{U}$. Note that $\mathbf{U}^1 = \mathbf{D}_V^1$ but the equality does not hold for larger $n$: each sample in $\mathbf{D}_V^n$ depends on $V$ but each sample in $\mathbf{U}^n$ is i.i.d.

We will give lower bounds that show no $(\varepsilon, \delta)$-internally private algorithm can distinguish $\mathbf{U}^n$ from $\mathbf{D}_V^n$. We will choose the family $\{\mathbf{D}_v\}$ so that any algorithm solving feature selection (or any of the other related problems) must distinguish $\mathbf{U}^n$ from $\mathbf{D}_V^n$, which is how we will obtain sample-complexity lower bounds.

For background, we briefly recap the way to use this setup to prove lower bounds under non-interactive local privacy. Here, one chooses the data from the mixture $\mathbf{D}_V^n$, and a lemma of Duchi, Jordan, and Wainwright [26] gives a bound on the mutual information between the output of the protocol $\mathcal{P}$ and the identity of the random mixture component $V$:

$$I(\mathcal{P}(\mathbf{D}_V^n); V) = O(n \cdot \varepsilon^2 \cdot \|\{\mathbf{D}_v\}\|_{\infty\to 2}^2) \tag{3.3}$$

where the $(\infty \to 2)$-norm[3]

$$\|\{\mathbf{D}_v\}\|_{\infty\to 2}^2 = \sup_{f:\mathcal{X}\to[\pm 1]} \mathop{\mathbb{E}}_{V}\left[\left(\mathop{\mathbb{E}}_{x\sim\mathbf{D}_V}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathbf{U}}[f(x)]\right)^2\right]$$

[3]We call this quantity the $(\infty\to 2)$-norm because it is equal to the better known $(\infty\to 2)$-norm, $\sup_z \|Mz\|_2/\|z\|_\infty$, of the matrix $M$ defined by $M_{v,x} = \mathbf{D}_v(x) - U(x)$.

is the crucial quantity determining how hard these distributions are to distinguish subject to local differential privacy. For intuition, note that this quantity satisfies the relationship

$$\|\{\mathbf{D}_v\}\|_{\infty\to 2}^2 \le \mathop{\mathbb{E}}_V\left[\sup_{f:\mathcal{X}\to[\pm 1]}\left(\mathop{\mathbb{E}}_{x\sim\mathbf{D}_V}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathbf{U}}[f(x)]\right)^2\right] = 4\cdot\mathop{\mathbb{E}}_V\left[\mathrm{d_{TV}}(\mathbf{D}_V,\mathbf{U})^2\right],$$

but it can be much smaller than $4\cdot\mathbb{E}_V(\mathrm{d_{TV}}(\mathbf{D}_V,\mathbf{U})^2)$, which is crucial for proving tight lower bounds.

Given this lemma, and a construction of a hard distribution family such that $\|\{\mathbf{D}_v\}\|_{\infty\to 2}^2$ is small, it is not hard to deduce a lower bound on the number of samples $n$ required to identify the specific mixture component $V$. It is also not too difficult to construct a family of hard distributions for all of our problems of interest (see Section 3.3.3).

With this state-of-affairs, it is tempting to argue that a mutual-information bound analogous to (3.3) holds for internally private algorithms. However, we can simulate the pointer-chasing result (Theorem 2.3.6) by an internally private algorithm; the implication is that the mutual information can be unbounded, showing that the purely information-theoretic approach used to prove lower bounds for the local model cannot work for internal privacy.[4]

Nonetheless, we prove the following indistinguishability lemma for any internally private $\mathcal{Q}$:

$$\mathrm{d_{TV}}(\mathcal{Q}(\mathbf{U}^n),\mathcal{Q}(\mathbf{D}_V^n)) \le O(n\cdot\varepsilon\cdot\|\{\mathbf{D}_v\}\|_{\infty\to 2}) \tag{3.4}$$

Although this bound is quantitatively somewhat weaker than (3.3)—in ways that are actually crucial to avoid proving false statements—it is nonetheless sufficient to give tight lower bounds for all of the problems we consider. The value of this lemma is that, even though the information-theoretic bounds that are used in the local model are false for the pan-private model, the exact same constructions of hard distributions can be used to obtain lower bounds for internal privacy! And in turn, we obtain lower bounds for pan-privacy and robust shuffle privacy.

The proof of (3.4) uses a hybrid argument, where we transition between data sampled from $\mathbf{U}^n$ and data sampled from $\mathbf{D}_V^n$. Namely, we fix a value of $i$ between 0 and $n$ and consider the case where the first $i$ inputs are sampled from $\mathbf{U}^i$ and the remaining $n-i$ inputs are sampled from $\mathbf{D}^{n-i}$. We then bound the total variation distance between the $i$-th case and the $(i+1)$-st case and apply the triangle inequality. In each step, we carefully argue that the total variation distance between the two cases follows from a careful application of (3.3) to the algorithm that computes the internal state after viewing the first $i$ inputs, which is why we ultimately get a bound of a similar form.

### 3.3.2 Main lower bound for internal privacy

Let $\mathcal{Q}$ be an $(\varepsilon,\delta)$-internally private algorithm. Let $\{\mathbf{D}_v\}_{v\in\mathcal{V}}$ be a family of distributions, $V$ be uniform over $\mathcal{V}$, and $\mathbf{U} = \mathop{\mathbb{E}}_V[\mathbf{D}_V]$ be the uniform mixture over the distributions.

The main goal of this section is to prove the following theorem.

**Theorem 3.3.2.** *If $\{\mathbf{D}_v\}_{v\in\mathcal{V}}$ is a family of distributions and $\mathcal{Q}$ is an $(\varepsilon,\delta)$-internally private algorithm such that* [5] *$\delta\log\frac{|\mathcal{V}|}{\delta} \ll \varepsilon^2\|\{\mathbf{D}_v\}\|_{\infty\to 2}^2$ and $\mathrm{d_{TV}}(\mathcal{Q}(\mathbf{D}_V^n),\mathcal{Q}(\mathbf{U}^n))$ is larger than a positive constant, then*

$$n \ge \Omega\left(\frac{1}{\varepsilon\|\{\mathbf{D}_v\}\|_{\infty\to 2}}\right)$$

*More generally, $n \ge 1/O(\varepsilon\|\{\mathbf{D}_v\}\|_{\infty\to 2} + \sqrt{\delta\log\frac{|\mathcal{V}|}{\delta}})$*

[4] $\mathcal{P}_{\mathsf{HIST}}$ crucially uses the full generality of $(\varepsilon,\delta)$-differential privacy for $\delta > 0$, however, even for stricter variants of differential privacy where the mutual information is bounded, we do not know how to obtain a mutual-information bound as strong as (3.3) for any of these variants.

[5] We use $x \ll y$ to indicate that $x \le cy$ for a sufficiently small numerical constant $c > 0$.

The main tool we use to prove Theorem 3.3.2 is the following information inequality.

**Lemma 3.3.3.** *For any $(\varepsilon, \delta)$-internally private algorithm $\mathcal{Q}$,*

$$\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}(\mathbf{D}_V^n), \mathcal{Q}(\mathbf{U}^n)) \le n \cdot \sqrt{\tfrac{1}{2} I_{\varepsilon,\delta}(\{\mathbf{D}_v\})}$$

*where we define* $I_{\varepsilon,\delta}(\{\mathbf{D}_v\}) = \sup_{\substack{\mathcal{Q}:\mathcal{X}\to\mathcal{Z} \\ (\varepsilon,\delta)\text{-DP}}} I(\mathcal{Q}(\mathbf{D}_V); V)$

*Proof of Lemma 3.3.3.* As a shorthand, let $\mathbf{B}_i$ denote the distribution of $\mathcal{Q}(\mathbf{U}^i, \mathbf{D}_V^{n-i})$. This is the distribution of the algorithm's output on a data stream where the first $i$ elements are sampled i.i.d. from $\mathbf{U}$ and the rest from $\mathbf{D}_V$. Note that $\mathbf{B}_0 = \mathcal{Q}(\mathbf{D}_V^n)$ and $\mathbf{B}_n = \mathcal{Q}(\mathbf{U}^n)$. By the triangle inequality we have

$$\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}(\mathbf{D}_V^n), \mathcal{Q}(\mathbf{U}^n)) = \mathrm{d}_{\mathrm{TV}}(\mathbf{B}_0, \mathbf{B}_n) \le \sum_{i=1}^{n} \mathrm{d}_{\mathrm{TV}}(\mathbf{B}_{i-1}, \mathbf{B}_i).$$

Thus, in order to prove the theorem it is enough to show that for every $i = 1, \ldots, n$,

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{B}_{i-1}, \mathbf{B}_i) \le \sqrt{\tfrac{1}{2} I_{\varepsilon,\delta}(\{\mathbf{D}_v\})} \tag{3.5}$$

Before proving (3.5), we give a simplified diagram of the relevant random variables in the two distributions $\mathbf{B}_{i-1}, \mathbf{B}_i$ in Figure 3.1. For the purposes of comparing $\mathbf{B}_{i-1}$ and $\mathbf{B}_i$, we can group all of the inputs $X_1, \ldots, X_{i-1} \sim \mathbf{U}^{i-1}$ into one random variable and all of the inputs $X_{i+1\cdots n} \sim \mathbf{D}_V^{n-i}$ into another random variable. Moreover, in $\mathbf{B}_{i-1}$, $X_i$ is drawn from $\mathbf{D}_V$, for the same choice of $V$ as $X_{i+1\cdots n}$, whereas in $\mathbf{B}_i$, $X_i$ is drawn from $\mathbf{U}$.

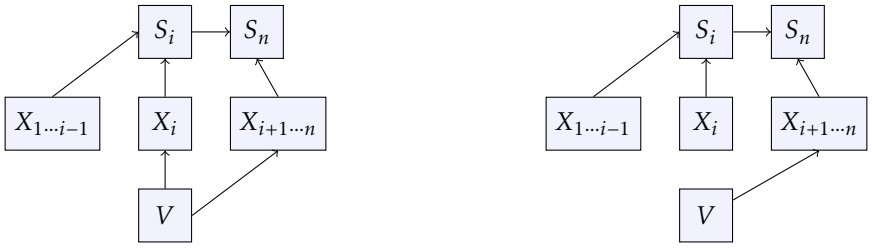


Figure 3.1: A simplified diagram of the relevant random variables in $\mathbf{B}_{i-1}$ (left) and $\mathbf{B}_i$ (right). $X_i$ is the $i$-th sample. $V$ determines the sampling distribution and the internal states $S_i, S_n$ are computed from the samples. Arrows indicate dependence.

Observe that the random variables $S_i$ and $X_{i+1\cdots n}$ have the same *marginal* distributions in both $\mathbf{B}_{i-1}, \mathbf{B}_i$. But the *joint* distributions are distinct: in $\mathbf{B}_{i-1}$, they are correlated by the shared choice of $V$ while in $\mathbf{B}_i$, they are independent. Moreover, $S_n$ is a post-processing of the pair $(S_i, X_{i+1\cdots n})$. Thus, using $(S_i, X_{i+1\cdots n})$ to denote the joint distribution of $S_i(V)$ and $X_{i+1\cdots n}(V)$ in $\mathbf{B}_{i-1}$, and applying the data-processing inequality, we have

$$\begin{aligned} \mathrm{d}_{\mathrm{TV}}(\mathbf{B}_{i-1}, \mathbf{B}_i) &\le \mathrm{d}_{\mathrm{TV}}((S_i, X_{i+1\cdots n}), (S_i \otimes X_{i+1\cdots n})) \\ &\le \mathop{\mathbb{E}}_{s_i \sim S_i} \left[ \mathrm{d}_{\mathrm{TV}}(X_{i+1\cdots n}|_{S_i = s_i}, X_{i+1\cdots n}) \right] \end{aligned} \tag{3.6}$$

where the last inequality uses the following fact.

**Fact 3.3.4.** *If $(A, B)$ and $(A, B')$ are joint distributions,* $\mathrm{d}_{\mathrm{TV}}((A, B), (A, B')) \le \mathop{\mathbb{E}}_{a \sim A} [\mathrm{d}_{\mathrm{TV}}(B|_{A=a}, B'|_{A=a})]$.

*Proof.* Given a set $T \subseteq \mathcal{A} \times \mathcal{B}$, define $T|_{A=a} = \{b : (a,b) \in T\}$. Then, we have

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}((A,B),(A,B')) \\
&= \sup_T \mathbb{P}[(A,B) \in T] - \mathbb{P}[(A,B') \in T] \\
&= \sup_T \mathop{\mathbb{E}}_{a\sim A}[\mathbb{P}[B|_{A=a} \in T|_{A=a}] - \mathbb{P}[B'|_{A=a} \in T|_{A=a}]] \\
&\le \sup_T \mathop{\mathbb{E}}_{a\sim A}[\mathrm{d}_{\mathrm{TV}}(B|_{A=a}, B'|_{A=a})] \\
&= \mathop{\mathbb{E}}_{a\sim A}[\mathrm{d}_{\mathrm{TV}}(B|_{A=a}, B'|_{A=a})]
\end{aligned}$$

This completes the proof. □

Next, since $S_i$ and $X_{i+1\cdots n}$ are independent conditioned on $V$, we have

$$\mathop{\mathbb{E}}_{s_i\sim S_i}\left[\mathrm{d}_{\mathrm{TV}}(X_{i+1\cdots n}|_{S_i=s_i}, X_{i+1\cdots n})\right] \le \mathop{\mathbb{E}}_{s_i\sim S_i}\left[\mathrm{d}_{\mathrm{TV}}(V|_{S_i=s_i}, V)\right] \tag{3.7}$$

where we use the following fact.

**Fact 3.3.5.** *If $(A,B,C)$ are jointly distributed random variables and $A$ and $B$ are independent conditioned on $C$, then for every $a \in \mathrm{supp}(A)$, $\mathrm{d}_{\mathrm{TV}}(B|_{A=a}, B) \le \mathrm{d}_{\mathrm{TV}}(C|_{A=a}, C)$.*

*Proof.* Let $T$ be an arbitrary subset of $\mathcal{B}$, then we have

$$\begin{aligned}
&\mathbb{P}[B \in T \mid A = a] - \mathbb{P}[B \in T] \\
&= \mathop{\mathbb{E}}_{c\sim C|_{A=a}}[\mathbb{P}[B \in T \mid A = a, C = c]] - \mathop{\mathbb{E}}_{c\sim C}[\mathbb{P}[B \in T \mid C = c]] \\
&= \mathop{\mathbb{E}}_{c\sim C|_{A=a}}[\mathbb{P}[B \in T \mid C = c]] - \mathop{\mathbb{E}}_{c\sim C}[\mathbb{P}[B \in T \mid C = c]] && \text{(conditional independence)} \\
&\le \sup_{f:\mathcal{C}\to[0,1]} \mathop{\mathbb{E}}_{c\sim C|_{A=a}}[f(c)] - \mathop{\mathbb{E}}_{c\sim C}[f(c)] \\
&= \mathrm{d}_{\mathrm{TV}}(C|_{A=a}, C)
\end{aligned}$$

where the final inequality is because $f(c) = \mathbb{P}[B \in T \mid C = c]$ is a function mapping $\mathcal{C} \to [0,1]$. Therefore we have

$$\mathrm{d}_{\mathrm{TV}}(B|_{A=a}, B) = \sup_T \mathbb{P}[B \in T \mid A = a] - \mathbb{P}[B \in T] \le \mathrm{d}_{\mathrm{TV}}(C|_{A=a}, C),$$

as desired. □

From this point we can calculate

$$\begin{aligned}
\mathop{\mathbb{E}}_{s_i\sim S_i}\left[\mathrm{d}_{\mathrm{TV}}(V|_{S_i=s_i}, V)\right] &\le \sqrt{\mathop{\mathbb{E}}_{s_i\sim S_i}\left[\mathrm{d}_{\mathrm{TV}}(V|_{S_i=s_i}, V)^2\right]} && \text{(Jensen's Inequality)} \\
&\le \sqrt{\mathop{\mathbb{E}}_{s_i\sim S_i}\left[\tfrac{1}{2}\cdot \mathrm{d}_{\mathrm{KL}}(V|_{S_i=s_i} \| V)\right]} && \text{(Pinsker's Inequality)} \\
&= \sqrt{\mathop{\mathbb{E}}_{s_i\sim S_i}\left[\tfrac{1}{2}\cdot \mathrm{d}_{\mathrm{KL}}((S_i,V) \| (S_i \otimes V))\right]} && \text{(chain rule for KL-divergence)} \\
&\le \sqrt{\tfrac{1}{2}\cdot I(S_i;V)} && \text{(definition of mutual information)}
\end{aligned}$$

Lastly, we argue that $I(S_i;V) \le I_{\varepsilon,\delta}(\{\mathbf{D}_v\})$ using privacy. The intuition is that privacy requires $S_i$ to be $(\varepsilon,\delta)$-differentially private as a function of the prefix $X_1,\ldots,X_i$. Moreover, $X_1,\ldots,X_{i-1}$ are drawn from the

fixed distribution $\mathbf{U}^{i-1}$ that is independent from $V$. Therefore, we can fix the distribution of $X_1, \ldots, X_{i-1}$ and view $S_i$ as an $(\varepsilon, \delta)$-differentially private algorithm of just $X_i$.

Specifically, given an $(\varepsilon, \delta)$-internally private algorithm $\mathcal{Q}$ and index $i \in [n]$, define the function $f_i : \mathcal{X} \to \mathcal{R}$ as follows: $f_i(x)$ samples $X_1, \ldots, X_{i-1} \sim \mathbf{U}^{i-1}$, computes $s_1 = \mathcal{Q}_1(X_1)$, $s_2 = \mathcal{Q}_2(X_2, s_1)$, $\ldots$, $s_{i-1} = \mathcal{Q}_{i-1}(X_{i-1}, s_{i-2})$, and outputs $r = \mathcal{Q}_i(x, s_{i-1})$. Internal privacy guarantees that $f_i(x) = \mathcal{Q}_i(X_1, \ldots, X_{i-1}, x)$ is $(\varepsilon, \delta)$-differentially private as a algorithm of $x$. Note that $S_i|_{X_i = x}$ is distributed identically as $f_i(x)$. Therefore,

$$\sqrt{\tfrac{1}{2} I(S_i; V)} = \sqrt{\tfrac{1}{2} I(\mathcal{Q}_i(\mathbf{D}_V); V)} \leq \sqrt{\tfrac{1}{2} I_{\varepsilon,\delta}(\{\mathbf{D}_v\})}$$

Combining with the previous calculations gives

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{B}_{i-1}, \mathbf{B}_i) \leq \sqrt{\tfrac{1}{2} I_{\varepsilon,\delta}(\{\mathbf{D}_v\})},$$

as desired. □

To use Lemma 3.3.3 we need a bound on the mutual information $I_{\varepsilon,\delta}(\{\mathbf{D}_v\})$. A result of Duchi, Jordan, and Wainwright [26], gives such a bound for the case of $\delta = 0$.

**Lemma 3.3.6** ([26]). $I_{\varepsilon,0}(\{\mathbf{D}_v\}) \leq O(\varepsilon^2 \|\{\mathbf{D}_v\}\|^2_{\infty\to 2})$.

We give a simple extension to the case of $\delta > 0$.

**Lemma 3.3.7.** $I_{\varepsilon,\delta}(\{\mathbf{D}_v\}) \leq O(\varepsilon^2 \|\{\mathbf{D}_v\}\|^2_{\infty\to 2} + \delta \log \frac{|\mathcal{V}|}{\delta})$.

Therefore, we will obtain Theorem 3.3.2 as an immediate corollary of Lemma 3.3.3 and Lemma 3.3.7. The proof of Lemma 3.3.7 from Lemma 3.3.6 relies on Lemma 1.3.10.

*Proof of Lemma 3.3.7.* Let $\mathcal{M}$ be any $(\varepsilon, \delta)$-differentially private function with input $x \in \mathcal{X}$. Lemma 1.3.10 guarantees that there exists a mechanism $\mathcal{Q}'$ that is $(2\varepsilon, 0)$-differentially private and satisfies

$$\forall x \in \mathcal{X} \;\; \mathrm{d}_{\mathrm{TV}}(\mathcal{M}(x), \mathcal{Q}'(x)) \leq \delta$$

In particular, $\mathrm{d}_{\mathrm{TV}}(\mathcal{M}(\mathbf{D}_V), \mathcal{Q}'(\mathbf{D}_V)) \leq \delta$. Therefore, there exists a joint distribution $(M, \mathcal{Q}')$ such that $\mathcal{M} = \mathcal{M}(\mathbf{D}_V)$, $\mathcal{Q}' = \mathcal{Q}'(\mathbf{D}_V)$ and $\mathbb{P}[M \neq \mathcal{Q}'] \leq \delta$. Let $B$ be the binary random variable $\mathbb{1}[\mathcal{M} \neq \mathcal{Q}']$. Thus, there is a joint distribution $(\mathcal{M}, \mathcal{Q}', B)$ such that $(B = 0 \implies R = R')$ and $\mathbb{P}[B \neq 0] \leq \delta$. Therefore,

$$\begin{aligned}
I(V; R) &\leq I(V; \mathcal{M}, \mathcal{Q}', B) \\
&\leq I(V; \mathcal{M}, \mathcal{Q}' \mid B) + H(B) \\
&= I(V; \mathcal{M}, \mathcal{Q}' \mid B = 0) \cdot \mathbb{P}[B = 0] + I(V; \mathcal{M}, \mathcal{Q}' \mid B = 1) \cdot \mathbb{P}[B = 1] + H(B) \\
&\leq I(V; \mathcal{Q}') + H(V)\delta + H(B) \\
&= I(V; \mathcal{Q}') + O(\delta \log |\mathcal{V}| + \delta \log(1/\delta)) \\
&\leq I_{2\varepsilon,0}(\{\mathbf{D}_v\}) + O(\delta \log |\mathcal{V}| + \delta \log(1/\delta)) \\
&= O(\varepsilon^2 \|\{\mathbf{D}_v\}\|^2_{\infty\to 2}) + O(\delta \log |\mathcal{V}| + \delta \log(1/\delta))
\end{aligned}$$

The lemma now follows by rewriting the final expression as $O(\delta \log \frac{|\mathcal{V}|}{\delta})$. □

### 3.3.3 A family of hard distributions

In order to apply Theorem 3.3.2 to a learning or optimization problem, we need a family of distributions $\{\mathbf{D}_v\}$ such that $\|\{\mathbf{D}_v\}\|_{\infty\to 2}$ is small and any accurate algorithm for the problem distinguishes $\mathbf{D}_V^n$ from $\mathbf{U}^n$. This subsection describes one such family we will use in most of our lower bound arguments. Throughout, we fix integer $d > 2$ and $\alpha \in (0, 1/2)$.

Let $\mathcal{X} = \{\pm 1\}^d$ be the data domain. For a non-empty set $\ell \subseteq [d]$ and a bit $b \in \{\pm 1\}^d$, we define the distribution $\mathbf{D}_{d,\ell,b,\alpha}$ to be uniform on $\{\pm 1\}^d$ except biased so that $\mathbb{E}_{x\sim\mathbf{D}_{d,\alpha,\ell,b,\alpha}}(\prod_{i\in t} x_i) = 2\alpha b$. Its probability mass function is

$$\mathbf{D}_{d,\ell,b,\alpha}(x) = \begin{cases} (1+2\alpha)2^{-d} & \text{if } \prod_{i\in t} x_i = b \\ (1-2\alpha)2^{-d} & \text{if } \prod_{i\in t} x_i = -b \end{cases} \tag{3.8}$$

Note that, by construction, for every non-empty $t' \neq t$, $\mathbb{E}_{x\sim\mathbf{D}_{d,\ell,b,\alpha}}(\prod_{i\in t'} x_i) = 0$.

For any positive integer $k \leq d$, we define the family

$$\mathcal{D}_{d,k,\alpha} = \{\mathbf{D}_{d,\ell,b,\alpha} : t \subseteq [d], |t| \in [k], b \in \{\pm 1\}\} \tag{3.9}$$

We also define the quantity $\binom{d}{\leq k} := \sum_{j=1}^{k} \binom{d}{j}$. The following two facts are immediate from the definition of $\mathcal{D}_{d,k,\alpha}$:

**Fact 3.3.8.** *The size of the family $\mathcal{D}_{d,k,\alpha}$ is $2 \cdot \binom{d}{\leq k}$.*

**Fact 3.3.9.** *The uniform mixture over the family $\mathcal{D}_{d,k,\alpha}$ is uniform over $\mathcal{X}$.*

The following lemma is implicit in many lower bounds for local differential privacy (e.g. [26, 57, 31]), although we reprove it here for completeness.

**Lemma 3.3.10.** $\|\mathcal{D}_{d,k,\alpha}\|_{\infty\to 2}^2 \leq 4\alpha^2/\binom{d}{\leq k}$

*Proof.* We begin by expanding the definition of the $(\infty \to 2)$ norm:

$$\begin{aligned}
\|\mathcal{D}_{d,k,\alpha}\|_{\infty\to 2}^2 &= \sup_{f:\mathcal{X}\to[\pm 1]} \sum_{\mathbf{D}\in\mathcal{D}_{d,k,\alpha}} \frac{1}{|\mathcal{D}_{d,k,\alpha}|} \cdot \left(\mathop{\mathbb{E}}_{x\sim\mathbf{D}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathbf{U}}[f(x)]\right)^2 \\
&= \sup_{f:\mathcal{X}\to[\pm 1]} \sum_{\substack{t\subseteq[d],|t|\in[k] \\ b\in\{\pm 1\}}} \frac{1}{|\mathcal{D}_{d,k,\alpha}|} \cdot \left(\sum_{x\in\{\pm 1\}^d} f(x)\cdot(\mathbf{D}_{d,\ell,b,\alpha}(x) - \mathbf{U}(x))\right)^2 \\
&= \sup_{f:\mathcal{X}\to[\pm 1]} \frac{1}{2\binom{d}{\leq k}} \cdot \sum_{\substack{t\subseteq[d],|t|\in[k] \\ b\in\{\pm 1\}}} \left(\sum_{x\in\{\pm 1\}^d} f(x)\cdot(\mathbf{D}_{d,\ell,b,\alpha}(x) - \mathbf{U}(x))\right)^2
\end{aligned} \tag{3.10}$$

The final equality comes from Fact 3.3.8. Note that (3.8) is equivalent to $\mathbf{D}_{d,\ell,b,\alpha}(x) = (1 + 2\alpha b \cdot \prod_{i\in t} x_i)2^{-d}$ and, via Fact 3.3.9, $\mathbf{U}(x) = 2^{-d}$. Thus,

$$\begin{aligned}
(3.10) &= \sup_{f:\mathcal{X}\to[\pm 1]} \frac{1}{2\binom{d}{\leq k}} \cdot \sum_{\substack{t\subseteq[d],|t|\in[k] \\ b\in\{\pm 1\}}} \left(\sum_{x\in\{\pm 1\}^d} f(x)\cdot 2\alpha b \cdot \prod_{i\in t} x_i \cdot 2^{-d}\right)^2 \\
&= \sup_{f:\mathcal{X}\to[\pm 1]} \frac{2\alpha^2}{\binom{d}{\leq k}} \cdot \sum_{\substack{t\subseteq[d],|t|\in[k] \\ b\in\{\pm 1\}}} \left(\sum_{x\in\{\pm 1\}^d} f(x)\cdot \prod_{i\in t} x_i \cdot 2^{-d}\right)^2 \\
&= \sup_{f:\mathcal{X}\to[\pm 1]} \frac{4\alpha^2}{\binom{d}{\leq k}} \cdot \sum_{t\subseteq[d],|t|\in[k]} \left(\sum_{x\in\{\pm 1\}^d} f(x)\cdot \prod_{i\in t} x_i \cdot 2^{-d}\right)^2 \\
&\leq \sup_{f:\mathcal{X}\to[\pm 1]} \frac{4\alpha^2}{\binom{d}{\leq k}} \cdot \sum_{t\subseteq[d]} \left(\sum_{x\in\{\pm 1\}^d} f(x)\cdot \prod_{i\in t} x_i \cdot 2^{-d}\right)^2
\end{aligned} \tag{3.11}$$

Define $\hat{f}(t) := \mathop{\mathbb{E}}_{X\sim\mathbf{U}}[f(X)\cdot\prod_{i\in t}X_i]$, the Fourier transform over the Boolean hypercube. This is precisely the term being squared above. So we have

$$\begin{aligned}(3.11) &= \frac{4\alpha^2}{\binom{d}{\leq k}}\cdot\sup_{f:\mathcal{X}\to[\pm1]}\sum_{t\subseteq[d]}\hat{f}(t)^2\\ &= \frac{4\alpha^2}{\binom{d}{\leq k}}\cdot\sup_{f:\mathcal{X}\to[\pm1]}\mathop{\mathbb{E}}_{X\sim\mathbf{U}}\left[f(X)^2\right] && \text{(Parseval's identity)}\\ &\leq \frac{4\alpha^2}{\binom{d}{\leq k}}\end{aligned}$$

This concludes the proof. □

The following is an immediate corollary of Theorem 3.3.2, Lemma 3.3.10, and Fact 3.3.8.

**Theorem 3.3.11.** *Let $\mathbf{D}_{d,L,B,2\alpha}$ denote a distribution chosen uniformly at random from $\mathcal{D}_{d,k,\alpha}$ (where $L$ is a uniformly random subset of $[d]$ with size $\leq k$ and $B$ is a uniformly random member of $\{\pm1\}$). If $M$ is an $(\varepsilon,\delta)$-internaly private algorithm such that $\delta\log\binom{d}{\leq k}/\delta \ll \alpha^2\varepsilon^2/\binom{d}{\leq k}$ and $\mathrm{d_{TV}}(M(\mathbf{D}^n_{d,L,B,\alpha}),M(\mathbf{U}^n))$ is larger than a positive constant, then*

$$n \geq \Omega\left(\frac{\sqrt{\binom{d}{\leq k}}}{\alpha\varepsilon}\right)$$

### 3.3.4 Lower bounds for feature selection

**Theorem 3.3.12.** *If $\mathcal{Q}=(\mathcal{Q}_1,\ldots,\mathcal{Q}_n,\mathcal{Q}_{\mathcal{O}})$ is an $(\varepsilon,\delta)$-internally private algorithm that solves $(\alpha,d)$-selection and $\delta\log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n=\Omega(\sqrt{d}/\alpha\varepsilon)$.*

*Proof.* Let $\mathbf{D}_{d,L,B,\alpha}$ denote a distribution chosen uniformly at random from $\mathcal{D}_{d,1,\alpha}$. We show that $\mathcal{Q}$ implies another $(\varepsilon,\delta)$-internally private algorithm $\mathcal{Q}'$ where the total variation distance between $\mathcal{Q}'(\mathbf{U}^n)$ and $\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha})$ is at least a positive constant.

Let $\mathbf{Rad}(\alpha)$ be the distribution over $\{\pm1\}$ with mean $\alpha$. For any $i\in[n]$, define $\mathcal{Q}'_i$ to be the internal update algorithm that does the following on input $x_i$:

1. Draw independent sample $Y_i$ from $\mathbf{Rad}(\alpha)$
2. $W_i \leftarrow (x_{i,1},x_{i,2},\ldots,x_{i,d},Y_i)$
3. Output $\mathcal{Q}_i(W_i,s_{i-1})$ if $i>1$ else $\mathcal{Q}_1(W_1)$

$\mathcal{Q}'$ is the online algorithm defined by $(\mathcal{Q}'_1,\ldots,\mathcal{Q}'_n,\mathcal{Q}_{\mathcal{O}})$. It is $(\varepsilon,\delta)$-internally private by virtue of using $\mathcal{Q}$, so it remains to lower bound the TV distance between $\mathcal{Q}'(\mathbf{U}^n)$ and $\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha})$.

$$\begin{aligned}&\mathrm{d_{TV}}(\mathcal{Q}'(\mathbf{U}^n),\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha}))\\ &\geq \mathbb{P}\left[\mathcal{Q}'(\mathbf{U}^n)=d+1\right]-\mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha})=d+1\right]\\ &= \mathbb{P}\left[\mathcal{Q}(\mathbf{D}^n_{d+1,\{d+1\},+1,\alpha/2})=d+1\right]-\mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha})=d+1\right] && (3.12)\\ &\geq \frac{99}{100}-\mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha})=d+1\right] && (3.13)\end{aligned}$$

To obtain (3.12), observe that $\mathcal{Q}'$ feeds into $\mathcal{Q}$ a stream of $n$ i.i.d. samples from a product distribution where the $(d+1)$-th coordinate has mean $\alpha$, while the rest have mean 0. In our notation, this product

distribution is $\mathbf{D}_{d+1,\{d+1\},+1,\alpha/2}$. Meanwhile, the inequality in (3.13) follows from the fact that $\mathcal{Q}$ solves $(\alpha, d+1)$-selection.

We now upper bound the probability in (3.13).

$$\begin{aligned}
&\mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha}) = d+1\right] \\
&= \mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha}) = d+1,\ B=-1\right] + \mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha}) = d+1,\ B=+1\right] \\
&\le \frac{1}{2} + \mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha}) = d+1,\ B=+1\right] \\
&= \frac{1}{2} + \sum_{j=1}^{d} \mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^n_{d,\{j\},+1,\alpha}) = d+1\right] \cdot \mathbb{P}\left[T=\{j\}, B=+1\right] \qquad (3.14)
\end{aligned}$$

We focus our attention on the first term in the product. Observe that $\mathcal{Q}'$ feeds to $\mathcal{Q}$ a stream of $n$ iid samples drawn from a distribution where coordinate $j \in [d]$ has mean $2\alpha$, coordinate $d+1$ has mean $\alpha$, and every other coordinate has mean 0. Here, $j$ is the correct answer to $(\alpha, d+1)$ selection; since $\mathcal{Q}$ solves $(\alpha, d+1)$-selection, $\mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^n_{d,\{j\},+1,\alpha}) = d+1\right] \le 1/100$. As a result,

$$(3.14) \le \frac{1}{2} + \sum_{j=1}^{d} \frac{1}{100} \cdot \mathbb{P}\left[L=\{j\}, B=+1\right] = \frac{1}{2} + \frac{1}{100} = \frac{51}{100}$$

Thus, $\mathrm{d_{TV}}(\mathcal{Q}'(\mathbf{U}^n), \mathcal{Q}'(\mathbf{D}^n_{d,L,B,\alpha})) \ge 99/100 - 51/100 = 12/25$. From Theorem 3.3.11, we conclude that

$$n = \Omega\left(\frac{1}{\varepsilon \|\mathcal{D}_{d,1,\alpha}\|_{\infty\to 2}}\right) = \Omega\left(\sqrt{d}/\alpha\varepsilon\right).$$

The claimed theorem now follows by rescaling $d$. □

We now adapt our proof to the robust shuffle privacy setting. For readability, we repeat the theorem statement from the beginning of the section:

**Theorem** (Restatement of Theorem 3.3.1)**.** If $\mathcal{P}$ is an $(\varepsilon, \delta)$-robustly shuffle private protocol that solves $(\alpha, d)$-selection and $\delta \log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.

*Proof.* As before, let $\mathbf{D}_{d,L,B,\alpha}$ denote a distribution chosen uniformly at random from $\mathcal{D}_{d,1,\alpha}$. Let $\mathcal{Q}_{\mathcal{P}}$ denote the $(O(\varepsilon), O(\delta))$-internally private algorithm given by Lemma 3.2.2. Like the preceding proof, we show that $\mathcal{Q}_{\mathcal{P}}$ implies an $(\varepsilon, \delta)$-internally private algorithm $\mathcal{Q}'$ that distinguishes between $\mathbf{U}^{n/2}$ and $\mathbf{D}^{n/2}_{d,L,B,\alpha}$. We construct $\mathcal{Q}'$ essentially identically, the differences being that we have $n/2$ instead of $n$ internal algorithms.

To bound the total variation distance between $\mathcal{Q}'(\mathbf{U}^{n/2})$ and $\mathcal{Q}'(\mathbf{D}^{n/2}_{d,L,B,\alpha})$ we follow the same steps as in the proof of Theorem 3.3.12 except we need to account for the reduction from robust shuffle privacy to internal privacy (Lemma 3.2.2)

$$\begin{aligned}
&\mathrm{d_{TV}}(\mathcal{Q}'(\mathbf{U}^{n/2}), \mathcal{Q}'(\mathbf{D}^{n/2}_{d,L,B,\alpha})) \\
&\ge \mathbb{P}\left[\mathcal{Q}'(\mathbf{U}^{n/2}) = d+1\right] - \mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^{n/2}_{d,L,B,\alpha}) = d+1\right] \\
&= \mathbb{P}\left[\mathcal{Q}(\mathbf{D}^{n/2}_{d+1,\{d+1\},+1,\alpha/2}) = d+1\right] - \mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^{n/2}_{d,L,B,\alpha}) = d+1\right] \\
&\ge \mathbb{P}\left[\mathcal{P}(\mathbf{D}^{n}_{d+1,\{d+1\},+1,\alpha/9}) = d+1\right] - \frac{1}{6} - \mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^{n/2}_{d,L,B,\alpha}) = d+1\right] && \text{(Lemma 3.2.2)} \\
&\ge \frac{99}{100} - \frac{1}{6} - \mathbb{P}\left[\mathcal{Q}'(\mathbf{D}^{n/2}_{d,L,B,\alpha}) = d+1\right] \\
&\ge \frac{99}{100} - \frac{1}{6} - \left(\frac{1}{2} + \frac{1}{100} + \frac{1}{6}\right) && \text{(Lemma 3.2.2)} \\
&= \frac{11}{75}
\end{aligned}$$

As before, we invoke Theorem 3.3.11 to conclude that $n = \Omega(\sqrt{d}/\alpha\varepsilon)$. The claimed theorem follows from rescaling $\alpha$ and $d$. □

### 3.3.5 Other lower bounds

Here, we use Theorem 3.3.2 and the same family $\mathcal{D}_{d,k,\alpha}$ to obtain lower bounds for other problems. We define these problems below and state the results; because the proofs have a repetitive structure, they are deferred to Appendix A.5.

**Definition 3.3.13** ($d$-Wise Simple Hypothesis Testing)**.** Let $d$ be any integer larger than 1 and let $\alpha$ be any real in the interval $(0, 1/2)$. An algorithm $\mathcal{M}$ solves *$d$-wise simple hypothesis testing with error $\alpha$ and sample complexity $n$* if, for any set of $d$ distributions $\mathcal{P}$ satisfying $\mathrm{d_{TV}}(\mathbf{D}, \mathbf{D}') \geq \alpha$ for every distinct pair $\mathbf{D}, \mathbf{D}' \in \mathcal{P}$, when given $n$ independent samples from an arbitrary $\mathbf{D} \in \mathcal{P}$ as input, the algorithm outputs $\mathbf{D}$ with probability $\geq 99/100$. This probability is over the randomness of the samples observed by $\mathcal{M}$ and $\mathcal{M}$ itself.

**Theorem 3.3.14.** *If $\mathcal{Q}$ is an $(\varepsilon, \delta)$-internally private algorithm that solves $d$-wise simple hypothesis testing with error $\alpha$ and $\delta \log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.*

**Theorem 3.3.15.** *If $\mathcal{P}$ is an $(\varepsilon, \delta)$-robustly shuffle private protocol that solves $d$-wise simple hypothesis testing with error $\alpha$ and $\delta \log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.*

**Definition 3.3.16.** Let $\alpha$ be any real in the interval $(0, 1/2)$ and let $k \leq d$ be any integers larger than 1. An algorithm $\mathcal{M}$ *solves $(d, k, \alpha)$-sparse mean estimation with sample complexity $n$* if, for any distribution $\mathbf{D}$ over $\{\pm 1\}^d$ whose mean $\vec{\mu}$ satisfies $\|\vec{\mu}\|_0 \leq k$, it receives $n$ independent samples from $\mathbf{D}$ as input and outputs $\vec{V} \in [-1, +1]^d$ such that $\|\vec{\mu} - \vec{V}\|_\infty \leq \alpha$ with probability at least 99/100. This probability is taken over the randomness of the samples observed by $\mathcal{M}$ and $\mathcal{M}$ itself.

**Theorem 3.3.17.** *If $\mathcal{Q}$ is an $(\varepsilon, \delta)$-internally private algorithm that solves $(d, 1, \alpha)$-sparse mean estimation and $\delta \log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.*

**Theorem 3.3.18.** *If $\mathcal{P}$ is an $(\varepsilon, \delta)$-robustly shuffle private protocol that solves $(d, 1, \alpha)$-sparse mean estimation and $\delta \log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.*

**Definition 3.3.19.** Let $\alpha$ be any real in the interval $(0, 1/2)$ and let $k \leq d$ be any integers larger than 1. An algorithm $\mathcal{M}$ *releases width-$k$ parities with error $\alpha$ and sample complexity $n$* if it takes $n$ independent samples from a distribution $\mathbf{D}$ over $\{\pm 1\}^d$ and reports a function $F : 2^{[d]} \to \mathbb{R}$ such that

$$\mathop{\mathbb{P}}_{\substack{\vec{X} \sim \mathbf{D}^n \\ F \sim \mathcal{M}(\vec{X})}} \left[ \forall \ell \subseteq [d], |\ell| \leq k \;\middle|\; F(\ell) - \mathop{\mathbb{E}}_{x \sim \mathbf{D}} \left[ \prod_{j \in \ell} x_j \right] \right| \leq \alpha \Bigg] \geq 99/100.$$

This probability is taken over the randomness of the samples observed by $\mathcal{M}$ and $\mathcal{M}$ itself.

**Theorem 3.3.20.** *If $\mathcal{Q}$ is an $(\varepsilon, \delta)$-internally private algorithm that releases width-$k$ parities with error $\alpha$ and $\delta \log \binom{d}{\leq k}/\delta \ll \alpha^2\varepsilon^2/\binom{d}{\leq k}$, then its sample complexity is $n = \Omega(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon)$.*

**Theorem 3.3.21.** *If $\mathcal{P}$ is an $(\varepsilon, \delta)$-robustly shuffle private protocol that releases width-$k$ parities with error $\alpha$ and $\delta \log \binom{d}{\leq k}/\delta \ll \alpha^2\varepsilon^2/\binom{d}{\leq k}$, then its sample complexity is $n = \Omega(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon)$.*

## 3.4 Parity Learning

In this section we give lower bounds on the sample complexity of (agnostic) parity learning. We define the domain to be $\mathcal{X} = \{\pm 1\}^{d+1}$ and interpret the bits at index $d+1$ to be labels of the strings. Our focus will be on signed parity functions: given a tuple $(\ell, b) \in 2^{[d]} \times \{\pm 1\}$ and a string $x \in \mathcal{X}$, we would like labels to predict the value $b \cdot \prod_{j \in \ell} x_j$. Specifically, for any distribution $\mathbf{D}$ over $\mathcal{X}$, we define error function

$$\mathrm{err}_{\mathbf{D}}(\ell, b) := \mathop{\mathbb{P}}_{X \sim \mathbf{D}}\left[ b \cdot \prod_{j \in \ell} X_j \neq X_{d+1} \right],$$

to be the probability of misclassifying a random test example.

**Definition 3.4.1.** Let $\alpha \in (0, 1/2)$ be a parameter and let $1 \leq k \leq d$ be integers. An algorithm $\mathcal{M}$ *learns width-$k$ signed parities with error $\alpha$ and sample complexity $n$* if it takes $n$ independent samples from a distribution $\mathbf{D}$ over $\mathcal{X}$ and reports a tuple $(L, B) \in 2^{[d]} \times \{\pm 1\}$ such that, with probability at least 99/100,

$$\mathrm{err}_{\mathbf{D}}(L, B) < \min_{\ell, b} \mathrm{err}_{\mathbf{D}}(\ell, b) + \alpha.$$

This probability is taken over the randomness of the samples observed by $\mathcal{M}$ and $\mathcal{M}$ itself.

**Theorem 3.4.2.** *If $\mathcal{Q}$ is an $(\varepsilon, \delta)$-internally private algorithm that learns width-$k$ signed parities with error $\alpha$ and $\delta \log\binom{d}{\leq k}/\delta \ll \alpha^2\varepsilon^2/\binom{d}{\leq k}$, then its sample complexity is $n = \Omega(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon)$.*

**Theorem 3.4.3.** *If $\mathcal{P}$ is an $(\varepsilon, \delta)$-robustly shuffle private protocol that learns width-$k$ signed parities with error $\alpha$ and $\delta \log\binom{d}{\leq k}/\delta \ll \alpha^2\varepsilon^2/\binom{d}{\leq k}$, then its sample complexity is $n = \Omega(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon)$.*

We will use the same technique developed for feature selection (Section 3.3), but this time combining Theorem 3.3.2 with a different family of distributions. For any $\alpha \in [0, 1/2]$ and any $\ell \subseteq [d]$, and a bit $b \in \{\pm 1\}$, we define the distribution $\mathbf{C}_{d,\ell,b,\alpha}$ to have probability mass function

$$\mathbf{C}_{d,\ell,b,\alpha}(x) = \begin{cases} (1+2\alpha)2^{-d-1} & \text{if } b \cdot \prod_{j \in \ell} x_j = x_{d+1} \\ (1-2\alpha)2^{-d-1} & \text{if } b \cdot \prod_{j \in \ell} x_j = -x_{d+1} \end{cases} \tag{3.15}$$

**Fact 3.4.4.** *For any $(\ell', b') \neq (\ell, b)$,*

$$\mathop{\mathbb{P}}_{X \sim \mathbf{C}_{d,\ell,b,\alpha}}\left[ b \cdot \prod_{j \in \ell} X_j = X_{d+1} \right] = \frac{1}{2} + \alpha$$

$$\mathop{\mathbb{P}}_{X \sim \mathbf{C}_{d,\ell,b,\alpha}}\left[ b' \cdot \prod_{j \in \ell'} X_j = X_{d+1} \right] \leq \frac{1}{2}$$

For dimension $d$, a parameter $k \leq d$, and $\alpha \in [0, 1/2]$, we define the family

$$\mathcal{C}_{d,k,\alpha} = \{\mathbf{C}_{d,\ell,b,\alpha} : \ell \subseteq [d], |\ell| \leq k, b \in \{\pm 1\}\} \tag{3.16}$$

The following facts about $\mathcal{C}_{d,k,\alpha}$ are straightforward to verify.

**Fact 3.4.5.** *The size of the family $\mathcal{C}_{d,k,\alpha}$ is $2\binom{d}{\leq k} + 2$.*

**Fact 3.4.6.** *The uniform mixture of the family $\mathcal{C}_{d,k,\alpha}$ is uniform over $\mathcal{X}$.*

We can also bound the $(\infty \to 2)$ norm.

**Lemma 3.4.7.** *For every $d \in \mathbb{N}$, $k \le d$, and $\alpha \in [0, 1/2]$,*

$$\|\mathcal{C}_{d,k,\alpha}\|_{\infty\to 2}^2 \le \frac{4\alpha^2}{\binom{d}{\le k}}$$

*Proof.* The proof proceeds almost identically with the proof of Lemma 3.3.10. Recall that we now take $\mathcal{X} = \{\pm 1\}^{d+1}$. We begin by expanding the definition of the $(\infty \to 2)$ norm:

$$\begin{aligned}
\|\mathcal{C}_{d,k,\alpha}\|_{\infty\to 2}^2 &= \sup_{f:\mathcal{X}\to[\pm 1]} \sum_{\mathbf{C}\in\mathcal{C}_{d,k,\alpha}} \frac{1}{|\mathcal{C}_{d,k,\alpha}|} \cdot \left(\mathop{\mathbb{E}}_{x\sim\mathbf{C}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathbf{U}}[f(x)]\right)^2 \\
&= \sup_{f:\mathcal{X}\to[\pm 1]} \sum_{\substack{t\subseteq[d],|t|\le k \\ b\in\{\pm 1\}}} \frac{1}{|\mathcal{C}_{d,k,\alpha}|} \cdot \left(\sum_{x\in\mathcal{X}} f(x)\cdot(\mathbf{C}_{d,t,b,\alpha}(x) - \mathbf{U}(x))\right)^2 \\
&= \sup_{f:\mathcal{X}\to[\pm 1]} \frac{1}{2\binom{d}{\le k}+2} \cdot \sum_{\substack{t\subseteq[d],|t|\le k \\ b\in\{\pm 1\}}} \left(\sum_{x\in\mathcal{X}} f(x)\cdot(\mathbf{C}_{d,t,b,\alpha}(x) - \mathbf{U}(x))\right)^2 \qquad (3.17)
\end{aligned}$$

The final equality comes from Fact 3.4.5. Note that (3.15) is equivalent to $\mathbf{C}_{d,t,b,\alpha}(x) = (1 + 2\alpha b \cdot \prod_{i\in t} x_i \cdot x_{d+1})2^{-d-1}$. We also have from Fact 3.4.6 that $\mathbf{U}(x) = 2^{-d-1}$. Thus,

$$\begin{aligned}
(3.17) &= \sup_{f:\mathcal{X}\to[\pm 1]} \frac{1}{2\binom{d}{\le k}+2} \cdot \sum_{\substack{t\subseteq[d],|t|\le k \\ b\in\{\pm 1\}}} \left(\sum_{x\in\mathcal{X}} f(x)\cdot 2\alpha b\cdot\prod_{i\in t} x_i \cdot 2^{-d-1}\right)^2 \\
&= \sup_{f:\mathcal{X}\to[\pm 1]} \frac{2\alpha^2}{\binom{d}{\le k}+1} \cdot \sum_{\substack{t\subseteq[d],|t|\le k \\ b\in\{\pm 1\}}} \left(\sum_{x\in\mathcal{X}} f(x)\cdot\prod_{i\in t} x_i \cdot 2^{-d-1}\right)^2 \\
&= \sup_{f:\mathcal{X}\to[\pm 1]} \frac{4\alpha^2}{\binom{d}{\le k}+1} \cdot \sum_{t\subseteq[d],|t|\le k} \left(\sum_{x\in\mathcal{X}} f(x)\cdot\prod_{i\in t} x_i \cdot 2^{-d-1}\right)^2 \\
&\le \sup_{f:\mathcal{X}\to[\pm 1]} \frac{4\alpha^2}{\binom{d}{\le k}} \cdot \sum_{t\subseteq[d]} \left(\sum_{x\in\mathcal{X}} f(x)\cdot\prod_{i\in t} x_i \cdot 2^{-d-1}\right)^2 \qquad (3.18)
\end{aligned}$$

Define $\hat{f}(t) := \mathop{\mathbb{E}}_{X\sim\mathbf{U}}[f(X)\cdot\prod_{i\in t} X_i]$, the Fourier transform over the Boolean hypercube. This is precisely the term being squared above. So we have

$$\begin{aligned}
(3.18) &= \frac{4\alpha^2}{\binom{d}{\le k}} \cdot \sup_{f:\mathcal{X}\to[\pm 1]} \sum_{t\subseteq[d]} \hat{f}(t)^2 \\
&= \frac{4\alpha^2}{\binom{d}{\le k}} \cdot \sup_{f:\mathcal{X}\to[\pm 1]} \mathop{\mathbb{E}}_{X\sim\mathbf{U}}\left[f(X)^2\right] && \text{(Parseval's identity)} \\
&\le \frac{4\alpha^2}{\binom{d}{\le k}}
\end{aligned}$$

This concludes the proof. □

We now prove our lower bound on internally private parity learning. To do so, we rely on Algorithm 16 which transforms an online algorithm for the learning problem into one that distinguishes distributions.

**Algorithm 16:** $\mathcal{Q}'$, an online algorithm

**Input:** Data stream $\vec{x} \in \mathcal{X}^m$; access to online algorithm $\mathcal{Q} : \mathcal{X}^n \to 2^{[d]} \times \{\pm 1\}$
**Output:** A random variable $Z \in \mathbb{R}$
$S_1 \leftarrow \mathcal{Q}_1(x_1)$
**For** $i \in [2, n]$
  $S_i \leftarrow \mathcal{Q}_i(x_i, S_{i-1})$
**For** $i \in [n+1, m]$
  **If** $i = n+1$ **:**
    $(\hat{L}, \hat{B}) \leftarrow \mathcal{Q}_{\mathcal{O}}(S_n)$
    $Z \sim \mathbf{Lap}(1/\varepsilon)$
  **Else**
    $(\hat{L}, \hat{B}, Z) \leftarrow S_{i-1}$
  **If** $\prod_{j \in \hat{L}} x_{i,j} = x_{i,d+1} \cdot \hat{B}$ **:**
    $Z \leftarrow Z + 1$
  $S_i \leftarrow (\hat{L}, \hat{B}, Z)$
**Return** $Z$

*Proof of Theorem 3.4.2.* Analogous to the proof of Theorem 3.3.12, let $\mathbf{C}_{d,L,B,\alpha}$ denote a distribution chosen uniformly at random from $\mathcal{C}_{d,k,\alpha}$. We argue that $\mathcal{Q}'$ is both $(\varepsilon, \delta)$-internally private and, when given $m = n + \Theta(1/\alpha\varepsilon)$ values from $\mathcal{X}$ as input, outputs a real number such that $\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}'(\mathbf{U}^m), \mathcal{Q}'(\mathbf{C}^m_{d,L,B,\alpha}))$ is larger than a constant. This allows invocation of Theorem 3.3.2.

At a high level, $\mathcal{Q}'$ has a training and a testing phase. In the training phase, it will execute $\mathcal{Q}$ on the first $n$ samples to obtain a signed parity function $(\hat{L}, \hat{B})$. In the testing phase, $\mathcal{Q}'$ will evaluate the function on the remaining samples and maintain an internally private estimate of the number of correct predictions. If the samples are drawn from the uniform mixture $\mathbf{U}$, then any choice of parity function makes a correct prediction with only $1/2$ probability (from Fact 3.4.4). But if the samples are drawn from any distribution $\mathbf{C}_{d,\ell,b,\alpha} \in \mathcal{C}_{d,k,\alpha}$, we know that $(\hat{L}, \hat{B}) = (\ell, b)$ with $\geq 99/100$ probability; conditioned on this event, our predictions will be correct with probability $1/2 + \alpha$ (again from Fact 3.4.4). Thus, the count of correct predictions will reliably differentiate between the two input cases.

Privacy: We will first prove privacy for user $i$ and intrusion time $t$. Specifically, we consider two arbitrary streams $\vec{x}, \vec{x}' \in \mathcal{X}^m$ that differ on index $i$. If $i \leq n$, $(\varepsilon, \delta)$-internal privacy follows immediately from the $(\varepsilon, \delta)$-internal privacy of $\mathcal{Q}$: the state observed by the adversary either precedes $i$ ($t < i$) or is a post-processing of a $(\varepsilon, \delta)$-private algorithm ($t \geq i$).

Otherwise, the state observed by the adversary either precedes $i$ or is a tuple $(\hat{L}, \hat{B}, Z)$, where the first two elements are independent of user $i$ and the third is distributed as $\sum_{u=n+1}^{t} \mathbb{1}\left[\prod_{j \in \hat{L}} x_{u,j} = x_{u,d+1} \cdot \hat{B}\right] + \mathbf{Lap}(1/\varepsilon)$. Because the summation is 1-sensitive, we obtain $\varepsilon$-differential privacy from the Laplace mechanism (Lemma 1.3.6).

Bound on TV distance: Now we show that the total variation distance between $\mathcal{Q}'(\mathbf{U}^m)$ and $\mathcal{Q}'(\mathbf{C}^m_{d,L,B,\alpha})$

is larger than a constant. Notice that, for any $\tau \in \mathbb{R}$,

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}'(\mathbf{C}^m_{d,L,B,\alpha}), \mathcal{Q}'(\mathbf{U}^m)) \\
&\geq \mathbb{P}\left[\mathcal{Q}'(\mathbf{C}^m_{d,L,B,\alpha}) > \tau\right] - \mathbb{P}\left[\mathcal{Q}'(\mathbf{U}^m) > \tau\right] \\
&= \left( \sum_{\substack{\ell \subseteq [d], |\ell| \leq k \\ b \in \{\pm 1\}}} \mathbb{P}\left[\mathcal{Q}'(\mathbf{C}^m_{d,\ell,b,\alpha}) > \tau\right] \cdot \mathbb{P}\left[(L,B) = (\ell,b)\right] \right) - \mathbb{P}\left[\mathcal{Q}'(\mathbf{U}^m) > \tau\right] \\
&= \left( \sum_{\substack{\ell \subseteq [d], |\ell| \leq k \\ b \in \{\pm 1\}}} \mathbb{P}\left[\mathcal{Q}'(\mathbf{C}^m_{d,\ell,b,\alpha}) > \tau \mid (\hat{L},\hat{B}) = (\ell,b)\right] \cdot \mathbb{P}\left[(\hat{L},\hat{B}) = (\ell,b)\right] \cdot \mathbb{P}\left[(L,B) = (\ell,b)\right] \right) - \mathbb{P}\left[\mathcal{Q}'(\mathbf{U}^m) > \tau\right] \\
&\geq \left( \sum_{\substack{\ell \subseteq [d], |\ell| \leq k \\ b \in \{\pm 1\}}} \mathbb{P}\left[\mathcal{Q}'(\mathbf{C}^m_{d,\ell,b,\alpha}) > \tau \mid (\hat{L},\hat{B}) = (\ell,b)\right] \cdot \frac{99}{100} \cdot \mathbb{P}\left[(L,B) = (\ell,b)\right] \right) - \mathbb{P}\left[\mathcal{Q}'(\mathbf{U}^m) > \tau\right] \qquad (3.19)
\end{aligned}$$

(3.19) comes from the fact that $\mathcal{Q}$ learns parities. Notice that, conditioned on $(\hat{L},\hat{B}) = (\ell,b)$, Fact 3.4.4 implies $\mathcal{Q}'(\mathbf{C}^m_{d,\ell,b,\alpha})$ is a sample from the convolution $\mathbf{Bin}(m-n, 1/2+\alpha) + \mathbf{Lap}(1/\varepsilon)$ with probability $\geq 99/100$.

Meanwhile, note that the equality $\mathbb{P}_{X \sim \mathbf{U}}\left[\prod_{j \in \ell} X_j = X_{d+1} \cdot b\right] = 1/2$ holds for any parity function $(\ell, b)$. Consequently, the output of the algorithm $\mathcal{Q}'(\mathbf{U}^m)$ is a sample from the convolution $\mathbf{Bin}(m-n, 1/2) + \mathbf{Lap}(1/\varepsilon)$.

Because $m - n = \Theta(1/\alpha\varepsilon)$, we can use a Chernoff bound to argue that there is some $\tau$ where

$$\begin{aligned}
(3.19) &\geq \left( \sum_{\substack{\ell \subseteq [d], |\ell| \leq k \\ b \in \{\pm 1\}}} \frac{99}{100} \cdot \frac{99}{100} \cdot \mathbb{P}\left[(L,B) = (\ell,b)\right] \right) - \frac{1}{100} \\
&= \frac{99^2 - 100}{10000}
\end{aligned}$$

Lemma 3.4.7 and Theorem 3.3.2 imply $m = \Omega\left(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon\right)$ and, in turn, $n = \Omega\left(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon\right)$. □

We conclude the section by proving our lower bound on robustly shuffle private parity learning.

*Proof of Theorem 3.4.3.* We repeat the construction, this time building $\mathcal{Q}'$ from $\mathcal{Q}_{\mathcal{P}}$ (the online algorithm derived from $\mathcal{P}$ via Lemma 3.2.2). To prove privacy of $\mathcal{Q}'$, we follow the same steps as in the proof of Theorem 3.4.2; we do not replicate the text here.

Lower bounding the total variation distance between $\mathcal{Q}'(\mathbf{U}^m)$ and $\mathcal{Q}'(\mathbf{C}^n_{d,L,B,\alpha})$ is also very similar

though we do have to account for the change from $\mathcal{P}$ to $\mathcal{Q}_{\mathcal{P}}$:

$$
\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}'(\mathbf{C}^m_{d,L,B,\alpha}), \mathcal{Q}'(\mathbf{U}^m)) \\
&= \left( \sum_{\substack{\ell \subseteq [d], |\ell| \le k \\ b \in \{\pm 1\}}} \mathbb{P}\left[\mathcal{Q}'(\mathbf{C}^m_{d,\ell,b,\alpha}) > \tau \mid (\hat{L}, \hat{B}) = (\ell, b)\right] \cdot \mathbb{P}\left[(\hat{L}, \hat{B}) = (\ell, b)\right] \cdot \mathbb{P}\left[(L, B) = (\ell, b)\right] \right) - \mathbb{P}\left[\mathcal{Q}'(\mathbf{U}^m) > \tau\right] \\
&\ge \left( \sum_{\substack{\ell \subseteq [d], |\ell| \le k \\ b \in \{\pm 1\}}} \mathbb{P}\left[\mathcal{Q}'(\mathbf{C}^m_{d,\ell,b,\alpha}) > \tau \mid (\hat{L}, \hat{B}) = (\ell, b)\right] \cdot \left(\frac{99}{100} - \frac{1}{6}\right) \cdot \mathbb{P}\left[(L, B) = (\ell, b)\right] \right) - \mathbb{P}\left[\mathcal{Q}'(\mathbf{U}^m) > \tau\right] \\
&\ge \left( \sum_{\substack{\ell \subseteq [d], |\ell| \le k \\ b \in \{\pm 1\}}} \frac{99}{100} \cdot \frac{247}{300} \cdot \mathbb{P}\left[(L, B) = (\ell, b)\right] \right) - \frac{1}{100} \\
&= \frac{99}{100} \cdot \frac{247}{300} - \frac{1}{100} = \frac{8051}{10000}
\end{aligned}
$$

Lemma 3.4.7 and Theorem 3.3.2 imply $m = \Omega\left(\sqrt{\binom{d}{\le k}}/\alpha\varepsilon\right)$ and, in turn, $n = \Omega\left(\sqrt{\binom{d}{\le k}}/\alpha\varepsilon\right)$. □

# Chapter 4

# Single-Message Shuffle Privacy

The earlier chapters focused on shuffle protocols that satisfy the constraint of robust differential privacy. This chapter focuses on protocols that limit the communication of each user to one message.

## 4.1 Binary Sums via Randomized Response

*Randomized response* is a straightforward distributed protocol for binary sums. It originates from work dating back to Warner in 1965 [60]. Here, each user $i$ reports a single message bit $y_i \leftarrow \mathcal{R}_{\mathsf{RR}}(x_i)$ whose bias encodes $x_i \in \{0,1\}$. We detail the randomizer and analyzer below. $p \in (0,1)$ is a public parameter which determines the level of noise in each bit.

$$\mathcal{R}_{\mathsf{RR}}(x) := \begin{cases} \mathbf{Ber}(1/2) & \text{with probability } p \\ x & \text{otherwise} \end{cases}$$

$$\mathcal{A}_{\mathsf{RR}}(\vec{y}) := \sum_{i=1}^{n} \frac{1}{1-p} \cdot (y_i - p/2)$$

Before analyzing randomized response as a shuffle protocol, we first interpret it as local protocol. Specifically, we show it is necessary to choose a large value of $p$ (near $1/2$) to achieve $(\varepsilon, \delta)$-local privacy.

**Theorem 4.1.1.** *If $\mathcal{R}_{\mathsf{RR}}$ is $(\varepsilon, \delta)$-differentially private, then $p \geq \frac{2(1-\delta)}{e^{\varepsilon}+1}$*

*Proof.* We will first obtain the probability that $\mathcal{R}_{\mathsf{RR}}$ outputs 1 on input 1, and then on input 0.

$$\begin{aligned} \mathbb{P}[\mathcal{R}_{\mathsf{RR}}(1) = 1] &= p \cdot 1/2 + (1-p) \\ &= 1 - p/2 \\ \mathbb{P}[\mathcal{R}_{\mathsf{RR}}(0) = 1] &= p \cdot 1/2 \\ &= p/2 \end{aligned}$$

Due to our privacy constraint, it must be the case that

$$\begin{aligned} \mathbb{P}[\mathcal{R}_{\mathsf{RR}}(1) = 1] &\leq e^{\varepsilon} \cdot \mathbb{P}[\mathcal{R}_{\mathsf{RR}}(0) = 1] + \delta \\ 1 - p/2 &\leq e^{\varepsilon} \cdot (p/2) + \delta \\ p &\geq \frac{2(1-\delta)}{e^{\varepsilon}+1} \end{aligned}$$

This concludes the proof. □

Due to this lower bound on $p$, the variance of the estimate reported by $\mathcal{A}_{\mathsf{RR}}$ must be linear in the number of users. This matches impossibility results by Beimel, Nissim, and Omri [14] and Chan, Shi, and Song [21].

But when we change the objective from local privacy to robust shuffle privacy, much greater accuracy is possible.

**Theorem 4.1.2.** *For $\varepsilon \le 1$, sufficiently small $\delta$, and $n = \Omega(\frac{1}{\varepsilon}\ln\frac{1}{\delta})$, there exists a choice of parameter $p \in (0,1)$ such that randomized response $\mathcal{P}_{\mathsf{RR}} = (\mathcal{R}_{\mathsf{RR}}, \mathcal{A}_{\mathsf{RR}})$ satisfies $(\varepsilon,\delta)$-robust shuffle privacy and estimates binary sums up to error*

$$O\left(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}}\right)$$

*with probability $\ge 99/100$.*

For comparison, $\mathcal{P}_{\mathsf{SYM}}$ achieves the same asymptotic error without having a lower bound on $n$ (Theorem 2.1.4). On the other hand, there is no bound on the maximum number of messages a user in $\mathcal{P}_{\mathsf{SYM}}$ will send.

Our first step in proving Theorem 4.1.2 is stating the error of the protocol in terms of $p$. As usual, the accuracy analysis is done under the assumption that all users obey the protocol ($\gamma = 1$).

**Claim 4.1.3** (Accuracy of $\mathcal{P}_{\mathsf{RR}}$)**.** *For any $n \in \mathbb{N}$, $x \in \{0,1\}^n$, and $p,\beta \in (0,1)$ such that $p > \frac{4}{n}\ln\frac{2}{\beta}$, the protocol $\mathcal{P}_{\mathsf{RR}} = (\mathcal{R}_{\mathsf{RR}}, \mathcal{A}_{\mathsf{RR}})$ estimates the sum with error behaving as follows:*

$$\mathbb{P}\left[\left|\mathcal{P}_{\mathsf{RR}}(x) - \sum_i x_i\right| > \sqrt{2np\ln\frac{2}{\beta}}\cdot\left(\frac{1}{1-p}\right)\right] \le \beta$$

*Proof.* We first show each term in the analyzer's summation, $\frac{1}{1-p}\cdot(y_i - p/2)$, is an unbiased estimate of $x_i$:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{1-p}\cdot(y_i - p/2)\right] &= \frac{1}{1-p}\cdot(\mathbb{E}[y_i] - p/2)\\ &= \frac{1}{1-p}\cdot((p\cdot 1/2 + (1-p)\cdot x_i) - p/2)\\ &= x_i\end{aligned}$$

We now derive the variance of each term in the summation:

$$\begin{aligned}\mathrm{Var}\left[\frac{1}{1-p}\cdot(y_i - p/2)\right] &= \left(\frac{1}{1-p}\right)^2\cdot\mathrm{Var}[y_i]\\ &= \left(\frac{1}{1-p}\right)^2\cdot\frac{p}{2}\cdot\left(1-\frac{p}{2}\right)\end{aligned}$$

Each term in the sum is an independent random variable with mean $x_i$, variance $\sigma^2 = \left(\frac{1}{1-p}\right)^2\cdot\frac{p}{2}\cdot\left(1-\frac{p}{2}\right)$, and magnitude at most $m = \frac{1}{1-p}$. Because $p > \frac{4}{n}\ln\frac{2}{\beta}$, we have that $\frac{n\sigma^2}{m^2} > \ln\frac{2}{\beta}$: an additive Chernoff bound implies the claimed inequality. □

The next step in proving Theorem 4.1.2 is to analyze privacy guarantees of the protocol.

**Claim 4.1.4** (Robust Shuffle Privacy of $\mathcal{P}_{\mathsf{RR}}$)**.** *Fix $\varepsilon \le 1$ and $\delta < 4e^{-9}$. If $n > \frac{208}{\varepsilon^2}\ln\frac{4}{\delta}$ and we assign $p \leftarrow \frac{104}{\varepsilon^2 n}\cdot\ln\frac{4}{\delta}$, then $\mathcal{P}_{\mathsf{RR}}$ is $(\varepsilon/\sqrt{\gamma},\delta)$-robustly shuffle private. If $\frac{208}{\varepsilon}\ln\frac{4}{\delta} \le n \le \frac{208}{\varepsilon^2}\ln\frac{4}{\delta}$ and $p \leftarrow 1 - \sqrt{\frac{\varepsilon^2 n}{832\ln(4/\delta)}}$, then $\mathcal{P}_{\mathsf{RR}}$ is $(\varepsilon/\gamma,\delta)$-robustly shuffle private.*

To see how the error bound in Theorem 4.1.2 follows from the above choice of $p$, consider the two parameter regimes:

1. When $\varepsilon \gg 1/\sqrt{n}$ then $p \approx \frac{1}{\varepsilon^2 n}\sqrt{\ln(1/\delta)} \ll n$, so the bound in Claim 4.1.3 is $O(\sqrt{np\ln(1/\beta)})$, which yields the desired bound.

2. When $\varepsilon \ll 1/\sqrt{n}$ then $1-p \approx \varepsilon\sqrt{n}/\sqrt{\ln(1/\delta)} \ll n$, so the bound in Claim 4.1.3 is $O\left(\frac{\sqrt{n\ln(1/\beta)}}{1-p}\right)$, which yields the desired bound.

In the remainder of this section, we will prove Claim 4.1.4. Our work will proceed in two stages. First, we will express the privacy guarantees of the protocol as a function of $p$ and honest fraction $\gamma$. Then we choose $p$ such that honest users are guaranteed $(\varepsilon,\delta)$-differential privacy when $\gamma = 1$. When $\gamma < 1$, the $\varepsilon$ privacy parameter will grow by a factor of either $1/\gamma$ or $1/\sqrt{\gamma}$, depending on its initial size.

### 4.1.1 Privacy as a function of $p$

Our goal will be to prove the following claim:

**Claim 4.1.5.** *For any $\delta < 4e^{-9}$, $\gamma n > 52\ln\frac{4}{\delta}$, and $\min(p,1-p) \geq \frac{52}{\gamma n}\ln\frac{4}{\delta}$, the algorithm $(\mathcal{S}\circ\mathcal{R}_{\mathsf{RR}}^{\gamma n})$ is $(\tilde{\varepsilon}(p,\gamma),\delta)$-differentially private, where*

$$\tilde{\varepsilon}(p,\gamma) = \sqrt{\frac{52\ln\frac{4}{\delta}}{\gamma np}}\cdot\left(1-p+2\sqrt{\frac{p\ln\frac{4}{\delta}}{\gamma n}}\right)$$

*Proof.* We will borrow some elements from the proof of Claim 2.1.6. Recall Lemma 1.3.16: proving that $\mathcal{M}_{\gamma n,\mathsf{RR}}$ is $(\tilde{\varepsilon},\delta)$-differentially private for all $\gamma \geq \tau$ will imply that $\mathcal{P}_{\mathsf{RR}}$ is $(\tilde{\varepsilon},\delta,\tau)$-robustly shuffle private for $n$ users. And recall that $\mathcal{M}_{\gamma n,\mathsf{RR}}$ is the algorithm that, on input $x_1,\ldots,x_{\gamma n}$, outputs the histogram which counts the occurrences of $\{0,1\}$ as produced by $\mathcal{R}_{\mathsf{RR}}(x_1),\ldots,\mathcal{R}_{\mathsf{RR}}(x_{\gamma n})$.

Because the number of messages is $\gamma n$, the frequency of 0 is computable from the frequency of 1: if $h_0, h_1$ count zeroes and ones, respectively, then $h_0 = \gamma n - h_1$. By post-processing (Fact 1.3.5), it suffices to prove that the count of ones as produced by $\mathcal{R}_{\mathsf{RR}}(x_1),\ldots,\mathcal{R}_{\mathsf{RR}}(x_{\gamma n})$ is an $(\tilde{\varepsilon}(p,\gamma),\delta)$-differentially private algorithm.

**Algorithm 17:** $\mathcal{M}_{m,p}(x_1\ldots x_m)$

**Input:** $(x_1\ldots x_{\gamma n}) \in \{0,1\}^m$, parameter $p \in (0,1)$.
**Output:** $y \in \{0,1,2,\ldots,m\}$

Sample $s \sim \mathbf{Bin}(m,p)$
Define $\mathcal{H}_s = \{H \subseteq [m] : |H| = s\}$ and choose $H \in \mathcal{H}_s$ uniformly at random
**Return** $y \leftarrow \sum_{i\notin H} x_i + \mathbf{Bin}\left(s,\frac{1}{2}\right)$

When $m = \gamma n$, we argue that $\mathcal{M}_{m,p}$ (Algorithm 17 above) is precisely that algorithm. In the execution of $\mathcal{R}_{\mathsf{RR}}(x_1),\ldots,\mathcal{R}_{\mathsf{RR}}(x_m)$, let $G$ denote the set of users who report $\mathbf{Ber}(1/2)$. Notice that $G$ is distributed identically with $H$ in $\mathcal{M}_p$: its size is $\mathbf{Bin}(\gamma n,p)$ and its members are uniformly random. Conditioning on $G = H$, this means the sum that is output by $\mathcal{M}_p(\vec{x})$ is $\sum_{i\in G}\mathbf{Ber}(1/2) + \sum_{i\notin G} x_i = \sum_{i\in H}\mathbf{Ber}(1/2) + \sum_{i\notin H}$, which is precisely the count of ones as produced by $\mathcal{R}_{\mathsf{RR}}(x_1),\ldots,\mathcal{R}_{\mathsf{RR}}(x_m)$.

Thus, our proof will be complete when we prove the following claim about $\mathcal{M}_{m,p}$:

**Claim 4.1.6.** *For any $\delta < 2e^{-9}$, $m > 52\ln\frac{2}{\delta}$, and $\min(p, 1-p) \geq \frac{52}{m}\ln\frac{2}{\delta}$, $\mathcal{M}_{m,p}$ is $(\varepsilon, 2\delta)$ differentially private, where*

$$\varepsilon = \sqrt{\frac{52\ln\frac{2}{\delta}}{mp}} \cdot \left(1 - p + 2\sqrt{\frac{p\ln\frac{2}{\delta}}{m}}\right)$$

□

To prove Claim 4.1.6, we first show that for *any* sufficiently large $H$, the final step (encapsulated by Algorithm 18) will ensure differential privacy for some parameters.

**Algorithm 18:** $\mathcal{M}_{m,H}$

**Input:** $(x_1 \dots x_m) \in \{0,1\}^m$, parameter $H \subseteq [m]$.
**Output:** $y_H \in \{0, 1, 2, \dots, m\}$

Sample $\eta \sim \mathbf{Bin}\left(|H|, \frac{1}{2}\right)$
**Return** $y_H \leftarrow \sum_{i \notin H} x_i + \eta$

**Claim 4.1.7.** *For any $\delta < 2e^{-9}$ and any $H \subseteq [m]$ where $|H| > 26\ln\frac{2}{\delta}$, $\mathcal{M}_{m,H}$ is $(\varepsilon, \delta)$-differentially private for*

$$\varepsilon = \ln\left(1 + \sqrt{\frac{26\ln\frac{2}{\delta}}{|H|}}\right) < \sqrt{\frac{26\ln\frac{2}{\delta}}{|H|}}$$

*Proof.* Notice that the function $f(\vec{x}) := \sum_{i \notin H} x_i$ is 1-sensitive, as changing one user's value between $\{0,1\}$ changes the sum by at most 1. Thus, the privacy guarantee immediately follows from Lemma 1.3.7. □

Next, we consider the case where $H$ is a *random* subset of $[m]$ with a *fixed* size $s$. For *any* sufficiently large value $s$ and *randomly* chosen $H$ where $|H| = s$, the privacy parameters improve significantly in the regime where $s$ is close to $n$. This is an *amplification via sampling argument* and an earlier example of the argument can be found in prior work by Kasiviswanathan et al. [49].

As with the addition of binomial noise, we treat the sampling of $H$ as the modular procedure Algorithm 19.

**Algorithm 19:** $\mathcal{M}_{m,s}$

**Input:** $(x_1, \dots, x_m) \in \{0,1\}^m$, parameter $s \in \{0, 1, 2, \dots, m\}$.
**Output:** $y_s \in \{0, 1, 2, \dots, m\}$

Define $\mathcal{H}_s = \{H \subseteq [m] : |H| = s\}$ and choose $H \leftarrow \mathcal{H}_s$ uniformly at random
**Return** $y_s \leftarrow \mathcal{M}_{m,H}(x)$

**Claim 4.1.8.** *For any $\delta < 2e^{-9}$ and any $26\ln\frac{2}{\delta} < s < m$, $\mathcal{M}_{m,s}$ is $(\varepsilon, \delta)$ differentially private for*

$$\varepsilon = \sqrt{\frac{26\ln\frac{2}{\delta}}{s}} \cdot \left(1 - \frac{s}{m}\right)$$

*Proof.* Fix $\vec{x} \sim \vec{x}' \in \{0,1\}^n$ where $x_j \neq x'_j$. $\mathcal{M}_{m,s}(\vec{x})$ selects $H$ uniformly from $\mathcal{H}_s$ and runs $\mathcal{M}_{m,H}(\vec{x})$; let $H$ denote the realization of $H$. To enhance readability, we will use the shorthand $\varepsilon_0(s) := \sqrt{\frac{26\log\frac{2}{\delta}}{s}}$. For any

$W \subset \{0, 1, 2, \ldots, m\}$, we aim to show that

$$\frac{\mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}) \in W] - \delta}{\mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}') \in W]} \leq \exp\left(\varepsilon_0(s) \cdot \left(1 - \frac{s}{n}\right)\right)$$

First, we have

$$\begin{aligned}
&\frac{\mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}) \in W] - \delta}{\mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}') \in W]} \\
&= \frac{\mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}) \in W,\ j \in H] + \mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}) \in W,\ j \notin H] - \delta}{\mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}') \in W,\ j \in H] + \mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}') \in W,\ j \notin H]} \\
&= \frac{\mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}) \in W \mid j \in H] \cdot \mathbb{P}[j \in H] + \mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}) \in W \mid j \notin H] \cdot \mathbb{P}[j \notin H] - \delta}{\mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}') \in W \mid j \in H] \cdot \mathbb{P}[j \in H] + \mathop{\mathbb{P}}_{H,\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}') \in W \mid j \notin H] \cdot \mathbb{P}[j \notin H]}
\end{aligned} \tag{4.1}$$

For brevity's sake, we will use the following shorthand

$$\begin{aligned}
q &:= \mathbb{P}[j \notin H] = (1 - s/m) \\
\tau(\vec{x}) &:= \mathop{\mathbb{P}}_{\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}) \in W \mid j \in H] \\
\zeta(\vec{x}) &:= \mathop{\mathbb{P}}_{\mathcal{M}_{m,H}}[\mathcal{M}_{m,H}(\vec{x}) \in W \mid j \notin H]
\end{aligned}$$

When user $j$ outputs a uniformly random bit, their private value has no impact on the distribution. This means $\tau(\vec{x}) = \tau(\vec{x}')$ and we therefore have

$$(4.1) = \frac{(1-q)\tau(\vec{x}) + q\zeta(\vec{x}) - \delta}{(1-q)\tau(\vec{x}) + q\zeta(\vec{x}')} \tag{4.2}$$

Since $s = |H|$ is sufficiently large, by Claim 4.1.7 we have $\zeta(\vec{x}) \leq (1 + \varepsilon_0(s)) \cdot \min\{\zeta(\vec{x}'), \tau(\vec{x})\} + \delta$.

$$\begin{aligned}
(4.2) &\leq \frac{(1-q)\tau(\vec{x}) + q \cdot (1 + \varepsilon_0(s)) \cdot \min\{\zeta(\vec{x}'), \tau(\vec{x})\} + \delta) - \delta}{(1-q)\tau(\vec{x}) + q\zeta(\vec{x}')} \\
&\leq \frac{(1-q)\tau(\vec{x}) + q \cdot (1 + \varepsilon_0(s)) \cdot \min\{\zeta(\vec{x}'), \tau(\vec{x})\}}{(1-q)\tau(\vec{x}) + q\zeta(\vec{x}')} \\
&= \frac{(1-q)\tau(\vec{x}) + q \cdot \min(\zeta(\vec{x}'), \tau(\vec{x})) + q \cdot \varepsilon_0(s) \cdot \min\{\zeta(\vec{x}'), \tau(\vec{x})\}}{(1-q)\tau(\vec{x}) + q\zeta(\vec{x}')} \\
&\leq \frac{(1-q)\tau(\vec{x}) + q\zeta(\vec{x}') + q \cdot \varepsilon_0(s) \cdot \min\{\zeta(\vec{x}'), \tau(\vec{x})\}}{(1-q)\tau(\vec{x}) + q\zeta(\vec{x}')} \\
&= 1 + \frac{q \cdot \varepsilon_0(s) \cdot \min\{\zeta(\vec{x}'), \tau(\vec{x})\}}{(1-q)\tau(\vec{x}) + q\zeta(\vec{x}')}
\end{aligned} \tag{4.3}$$

Observe that $\min\{\zeta(\vec{x}'), \tau(\vec{x})\} \le (1-q)\tau(\vec{x}) + q\zeta(\vec{x}')$, so

$$\begin{aligned}(4.3) &\le 1 + q \cdot \varepsilon_0(s) \\ &= 1 + \varepsilon_0(s) \cdot \left(1 - \frac{s}{m}\right) \\ &\le \exp\left(\varepsilon_0(s) \cdot \left(1 - \frac{s}{m}\right)\right) \\ &= \exp\left(\sqrt{\frac{26 \log \frac{2}{\delta}}{s}} \cdot \left(1 - \frac{s}{m}\right)\right)\end{aligned}$$

which completes the proof. □

Finally, we show that when the size $s$ is chosen *randomly* then $s$ is sufficiently large with high probability.

*Proof of Claim 4.1.6.* Given that $\min(p, 1-p) \ge \frac{52}{m} \ln \frac{2}{\delta}$, the variance of $s \sim \mathbf{Bin}(m, p)$ is sufficiently large to invoke an additive Chernoff bound. Specifically, $s \ge mp - 2\sqrt{mp(1-p)\ln \frac{2}{\delta}}$ with probability $1-\delta$. Note that this means

$$\begin{aligned}s &\ge mp - 2\sqrt{mp \ln \frac{2}{\delta}} \\ &\ge mp/2 && (mp \ge 16 \ln \tfrac{2}{\delta}) \\ &\ge 26 \ln \frac{2}{\delta} && (p \ge \tfrac{52}{m} \ln \tfrac{2}{\delta})\end{aligned}$$

For this range of $s$, we can invoke Claim 4.1.8. Our claim follows by substituting $s \ge mp/2$ into the term $\sqrt{\frac{26 \ln \frac{2}{\delta}}{s}}$ and then $s \ge mp - 2\sqrt{mp \ln \frac{2}{\delta}}$ into the term $1 - \frac{s}{m}$. □

### 4.1.2 Setting $p$ for target Privacy

Claim 4.1.5 tells us that $(\mathcal{S} \circ \mathcal{R}_{\mathsf{RR}}^{\gamma n})$ satisfies $(\tilde{\varepsilon}(p, \gamma), \delta)$-differential privacy, where

$$\tilde{\varepsilon}(p, \gamma) = \sqrt{\frac{52 \ln \frac{4}{\delta}}{\gamma n p}} \cdot \left(1 - p + 2\sqrt{\frac{p \ln \frac{4}{\delta}}{\gamma n}}\right)$$

Given the above bound, we now choose a value of $p$ that guarantees a target level of $(\varepsilon, \delta)$-differential privacy for the honest users when $\gamma = 1$. For any other $\gamma \in [1/2, 1]$, the privacy guarantee weakens by a factor of either $\sqrt{1/\gamma}$ or $1/\gamma$. We remark that we do not optimize for the constants.

**Claim** (Restatement of Claim 4.1.4)**.** Fix $\varepsilon \le 1$ and $\delta < 4e^{-9}$. If $n > \frac{208}{\varepsilon^2} \ln \frac{4}{\delta}$ and we assign $p \leftarrow \frac{104}{\varepsilon^2 n} \cdot \ln \frac{4}{\delta}$, then $\mathcal{P}_{\mathsf{RR}}$ is $(\varepsilon/\sqrt{\gamma}, \delta)$-robustly shuffle private.

If $\frac{208}{\varepsilon} \ln \frac{4}{\delta} \le n \le \frac{208}{\varepsilon^2} \ln \frac{4}{\delta}$ and $p \leftarrow 1 - \sqrt{\frac{\varepsilon^2 n}{832 \ln(4/\delta)}}$, then $\mathcal{P}_{\mathsf{RR}}$ is $(\varepsilon/\gamma, \delta)$-robustly shuffle private.

*Proof.* As suggested by the structure of the claim, the proof proceeds by case analysis.

Case 1: $n > \frac{208}{\varepsilon^2}\ln\frac{4}{\delta}$. If we could show that $\min(p,1-p) \geq \frac{52}{\gamma n}\ln\frac{4}{\delta}$, we can invoke Claim 4.1.5 to conclude that $\mathcal{P}_{\mathsf{RR}}$ satisfies $(\varepsilon/\sqrt{\gamma},\delta)$-robust shuffle privacy:

$$\begin{aligned}
\tilde{\varepsilon}(p,\gamma) &= \sqrt{\frac{52\ln\frac{4}{\delta}}{\gamma np}}\cdot\left(1-p+2\sqrt{\frac{p\ln\frac{4}{\delta}}{\gamma n}}\right)\\
&< \sqrt{\frac{52\ln\frac{4}{\delta}}{\gamma np}} && \text{(Bound on } \min(p,1-p)\text{)}\\
&< \varepsilon/\sqrt{\gamma} && \text{(Choice of } p\text{)}
\end{aligned}$$

It remains to prove $\min(p,1-p) \geq \frac{52}{\gamma n}\ln\frac{4}{\delta}$. The inequality $p \geq \frac{52}{\gamma n}\ln\frac{4}{\delta}$ is immediate from the fact that $\gamma \geq 1/2, \varepsilon \leq 1$. And we also have that $p \leq 1/2$ due to lower bound on $n$.

Case 2: $\frac{208}{\varepsilon}\ln\frac{4}{\delta} \leq n \leq \frac{208}{\varepsilon^2}\ln\frac{4}{\delta}$. In this regime, $p = 1-\sqrt{\frac{\varepsilon^2 n}{832\ln(4/\delta)}}$. Note that $p \geq 1/2$ due to the upper bound on $n$. Once again, if we could show that $\min(p,1-p) \geq \frac{52}{\gamma n}\ln\frac{4}{\delta}$, we can invoke Claim 4.1.5 to conclude that $\mathcal{P}_{\mathsf{RR}}$ satisfies $(\varepsilon/\gamma,\delta)$-robust shuffle privacy:

$$\begin{aligned}
\tilde{\varepsilon}(p,\gamma) &= \sqrt{\frac{52\ln\frac{4}{\delta}}{\gamma np}}\cdot\left(1-p+\sqrt{\frac{2p\ln\frac{4}{\delta}}{\gamma n}}\right)\\
&= \sqrt{\frac{52\ln\frac{4}{\delta}}{\gamma np}}\cdot\left(\sqrt{\frac{\varepsilon^2 n}{832\ln(4/\delta)}}+\sqrt{\frac{2p\ln\frac{4}{\delta}}{\gamma n}}\right) && \text{(Choice of } p\text{)}\\
&= \varepsilon/\sqrt{16p\gamma}+\frac{\sqrt{104}\ln\frac{4}{\delta}}{\gamma n}\\
&\leq \varepsilon/\sqrt{8\gamma}+\frac{\sqrt{104}\ln\frac{4}{\delta}}{\gamma n} && (p \geq 1/2)\\
&< \varepsilon/\gamma && (n > \tfrac{208}{\varepsilon}\ln\tfrac{4}{\delta})
\end{aligned}$$

It remains to prove $\min(p,1-p) \geq \frac{52}{\gamma n}\ln\frac{4}{\delta}$. Since $p \geq 1/2$, it will suffice to prove $\sqrt{\frac{\varepsilon^2 n}{832\ln(4/\delta)}} \geq \frac{52}{\gamma n}\ln\frac{4}{\delta}$.

$$\begin{aligned}
n &> \frac{208}{\varepsilon}\ln\frac{4}{\delta}\\
n^{3/2} &> \left(\frac{208}{\varepsilon}\ln\frac{4}{\delta}\right)^{3/2}\\
\varepsilon^{3/2}\cdot\sqrt{\frac{n}{832\ln(4/\delta)}} &\geq \frac{104}{n}\ln\frac{4}{\delta}\\
&\geq \frac{52}{\gamma n}\ln\frac{4}{\delta} && (\gamma \geq 1/2)
\end{aligned}$$

The proof is complete since $\varepsilon > \varepsilon^{3/2}$ for $0 < \varepsilon \leq 1$ □

## 4.2 The Limits of Single-Message Shuffle Privacy

In this section, we present a technique to obtain lower bounds for single-message shuffle protocols. Recall that these are the class of protocols where each user sends exactly one message to the shuffler. We use *removal lemmas* that give a structural characterization of such protocols: if we remove the shuffler of a single-message shuffle protocol, we are left with a local protocol whose privacy parameters can be

expressed in terms of the original protocol. We can therefore invoke local privacy lower bounds to obtain lower bounds for single-message shuffle privacy.

### 4.2.1 Pure differential privacy ($\delta = 0$)

The first removal lemma concerns pure differential privacy. It originated in joint work with Balcer [7].

**Lemma 4.2.1** (Pure D.P. Removal Lemma)**.** *If a single-message protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ satisfies pure shuffle privacy, then removing the shuffler leaves behind a pure locally private protocol. Specifically, $\mathcal{R}$ must satisfy $\varepsilon$-differential privacy on its own whenever $\mathcal{P}$ is $\varepsilon$-shuffle private.*

Looking back at Table 1.1, this result means every lower bound in the local privacy column can be adapted to single-message pure shuffle privacy without loss.

*Proof of Lemma 4.2.1.* Assume for contradiction that $\mathcal{R}$ is not $\varepsilon$-differentially private. So there are values $x, x' \in \mathcal{X}$ and a set $Y \subseteq \mathcal{Y}$ such that

$$\mathbb{P}\left[\mathcal{R}(x) \in Y\right] > e^{\varepsilon} \cdot \mathbb{P}\left[\mathcal{R}(x') \in Y\right].$$

Let $\vec{x} = (\underbrace{x, \ldots, x}_{n \text{ copies}})$ and $\vec{x}' = (x', \underbrace{x, \ldots, x}_{n-1 \text{ copies}})$. Now consider $Y^n$, the set of message vectors where each message belongs to $Y$.

$$\begin{aligned}
\mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}) \in Y^n\right] &= \mathbb{P}\left[\mathcal{R}^n(\vec{x}) \in Y^n\right] \\
&= \mathbb{P}\left[\mathcal{R}(x) \in Y\right]^n \\
&> e^{\varepsilon} \cdot \mathbb{P}\left[\mathcal{R}(x') \in Y\right] \cdot \mathbb{P}\left[\mathcal{R}(x) \in Y\right]^{n-1} \\
&= e^{\varepsilon} \cdot \mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}') \in Y^n\right]
\end{aligned}$$

which contradicts the fact that $(\mathcal{S} \circ \mathcal{R}^n)$ is $\varepsilon$-differentially private. □

### 4.2.2 Approximate differential privacy ($\delta > 0$)

To obtain lower bounds under approximate differential privacy, we must use a different lemma derived in joint work with Smith Ullman Zeber and Zhilyaev [23].

**Lemma 4.2.2** (Approximate D.P. Removal Lemma)**.** *If a single-message protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ satisfies $(\varepsilon, \delta)$-shuffle privacy for $n$ users, then $\mathcal{R}$ must satisfy $(\varepsilon + \ln n, \delta)$-differential privacy on its own.*

*Proof.* By assumption, $\mathcal{P}$ is $(\varepsilon, \delta)$-private. Let $\varepsilon_L$ be the supremum such that $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ is *not* $(\varepsilon_L, \delta)$-private. We will attempt to find a bound on $\varepsilon_L$; if $R$ is not $(\varepsilon_L, \delta)$-differentially private, there exist $Y \subset \mathcal{Y}$ and $x, x' \in \mathcal{X}$ such that

$$\mathbb{P}\left[\mathcal{R}(x') \in Y\right] > e^{\varepsilon_L} \cdot \mathbb{P}\left[\mathcal{R}(x) \in Y\right] + \delta$$

For brevity, define $p := \mathbb{P}\left[\mathcal{R}(x) \in Y\right]$ and $p' := \mathbb{P}\left[R(x') \in Y\right]$ so that we have

$$p' > e^{\varepsilon_L} \cdot p + \delta \tag{4.4}$$

We will show that if $\varepsilon_L$ is too large, then (4.4) will imply that $\mathcal{P}$ is *not* $(\varepsilon, \delta)$-differentially private, which contradicts our assumption. To this end, define the set $\mathcal{W} := \{W \in \mathcal{Y}^n \mid \exists i \; w_i \in Y\}$. Define two datasets $\vec{x} \sim \vec{x}'$ as

$$\vec{x} := (\underbrace{x, \ldots, x}_{n \text{ times}}) \quad \text{and} \quad \vec{x}' := (x', \underbrace{x, \ldots, x}_{n-1 \text{ times}})$$

Because $\mathcal{P}$ is $(\varepsilon, \delta)$-differentially private

$$\mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}') \in \mathcal{W}\right] \leq e^{\varepsilon} \cdot \mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}) \in W\right] + \delta_S \tag{4.5}$$

Now we have

$$\begin{aligned}
&\mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}) \in \mathcal{W}\right] \\
&= \mathbb{P}\left[\mathcal{S}(\underbrace{\mathcal{R}(x), \ldots, \mathcal{R}(x)}_{n \text{ times}}) \in \mathcal{W}\right] \\
&= \mathbb{P}\left[(\underbrace{\mathcal{R}(x), \ldots, \mathcal{R}(x)}_{n \text{ times}}) \in \mathcal{W}\right] && (\mathcal{W} \text{ is symmetric}) \\
&= \mathbb{P}\left[\exists i \;\; \mathcal{R}(x) \in Y\right] \leq n \cdot \mathbb{P}\left[\mathcal{R}(x) \in Y\right] && (\text{Union bound}) \\
&= np
\end{aligned}$$

where the second equality is because the set $W$ is closed under permutation, so we can remove the random permutation $\mathcal{S}$ without changing the probability. Similarly, we have

$$\begin{aligned}
\mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}') \in \mathcal{W}\right] = \mathbb{P}\left[(\mathcal{R}(x'), \underbrace{\mathcal{R}(x) \ldots, \mathcal{R}(x)}_{n-1 \text{ times}}) \in \mathcal{W}\right] \\
&\geq \mathbb{P}\left[\mathcal{R}(x') \in Y\right] = p' \\
&> e^{\varepsilon_L} \cdot p + \delta && (\text{By } (4.4))
\end{aligned}$$

Now, plugging the previous two inequalities into (4.5), we have

$$\begin{aligned}
e^{\varepsilon_L} \cdot p + \delta &< \mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}') \in \mathcal{W}\right] \\
&\leq e^{\varepsilon} \cdot \mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}) \in \mathcal{W}\right] \\
&\leq e^{\varepsilon} \cdot np + \delta
\end{aligned}$$

By rearranging and canceling terms in the above we obtain the conclusion

$$\varepsilon_L \leq \varepsilon + \ln n$$

Therefore $\mathcal{R}$ must satisfy $(\varepsilon + \ln n, \delta)$-differential privacy. □

Given the above lemma, we can invoke any lower bound that holds for $(\varepsilon + \ln n, \delta)$-local privacy. Ghazi et al. obtain such a lower bound for histograms and conclude the following:

**Theorem 4.2.3** (Ghazi et al. [38])**.** *Any single-message protocol that satisfies* $(1, o(1/n))$*-shuffle privacy and outputs histograms with* $\ell_\infty$ *error* $n/10$ *must have* $n = \Omega(\frac{\log k}{\log \log k})$.

In contrast, there is a central model algorithm where $n = O(1)$ suffices for the same privacy and accuracy regimes.

## 4.3 Optimality of Amplification Lemmas

In our randomized response protocol $\mathcal{P}_{\mathsf{RR}}$, recall that we chose a value of parameter $p$ in order to satisfy $(\varepsilon, \delta)$-shuffle privacy for $n$ users. When $\varepsilon = \Theta(1)$ and $n$ is sufficiently large, observe that $\mathcal{R}_{\mathsf{RR}}$ on its own offers $(\varepsilon_L, 0)$-privacy where $\exp(\varepsilon_L) = O(n/\log 1/\delta)$. In essence, the shuffling "amplifies" the poor but existing privacy guarantees of $\mathcal{R}_{\mathsf{RR}}$.

One might wonder whether the shuffler performs similar privacy amplification of other local randomizers. If so, we could easily construct shuffle protocols by simply making different parameter choices in existing locally private protocols. Balle et al. [11] and Erlingsson et al. [33] present so-called amplification-by-shuffling lemmas to answer this line of inquiry. Feldman, Talwar, and McMillan [35] derive the state-of-the-art amplification lemma and apply it to distribution estimation in $\ell_2$ norm, as well as stochastic gradient descent.

**Lemma 4.3.1** (From [35])**.** *Fix $\varepsilon > 0$, $\delta \in (0,1)$, and $n \in \mathbb{N}$. There exists a constant $\kappa$ such that if local randomizer $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ is $\varepsilon_L$-differentially private for $\varepsilon_L < \ln\left(\kappa \cdot \left(\frac{e^\varepsilon - 1}{e^\varepsilon + 1}\right)^2 \cdot \frac{n}{\ln(1/\delta)}\right)$, then any shuffle protocol using $\mathcal{R}$ is $(\varepsilon, \delta)$-differentially private for $n$ users.*

Observe that the above lemma demands an upper bound on $\varepsilon_L$(equivalently, $e^{\varepsilon_L}$). A natural line of inquiry is to determine how much this bound can be loosened, or if it is already optimal. Lemma 4.2.2 immediately implies that no amplification can guarantee $(\varepsilon, \delta)$-shuffle privacy for $n$ users if the randomizer does not satisfy $(\ln(e^\varepsilon n), \delta)$-local privacy. Combined with Lemma 4.3.1, we see that the optimal bound on $e^{\varepsilon_L}$ has to be linear in $n$.

We derive another bound on $\varepsilon_L$ by considering, again, the specific case of randomized response.

**Claim 4.3.2.** *For any $\varepsilon > 0$, $0 < \delta < 2/5$, and $n \in \mathbb{N}$, let $\varepsilon_L := \ln\left(2(e^\varepsilon + 1) \cdot \frac{n}{\ln(1/\delta)}\right)$. There exists a choice of parameter $p$ where the local randomizer $\mathcal{R}_{\mathsf{RR}}$ satisfies $\varepsilon_L$-local privacy but the shuffle protocol $\mathcal{P}_{\mathsf{RR}}$ does not satisfy $(\varepsilon, \delta)$-shuffle privacy for $n$ users. Consequently, no amplification lemma can obtain $(\varepsilon, \delta)$-shuffle privacy from a generic $\varepsilon_L$-locally private randomizer.*

This result means that the $\frac{n}{\ln(1/\delta)}$ term in Lemma 4.3.1 is necessary.

*Proof.* To streamline this proof, we will change the parameterization of randomized response. Specificially, we will use $q = p/2$ to denote the probability that $\mathcal{R}_{\mathsf{RR}}(x) = 1 - x$ (and $1 - q$ for the probability that $\mathcal{R}_{\mathsf{RR}}(x) = x$). Note that $\mathcal{R}_{\mathsf{RR}}$ satisfies $\ln \frac{1}{q}$-differential privacy on its own.

Consider the inputs $\vec{x} = (0, 0, \ldots, 0)$ and $\vec{x}' = (1, 0, \ldots, 0)$. Let $Y$ denote the event where the shuffler outputs $n$ copies of 0. From the independence of randomized response, observe that

$$\mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}_{\mathsf{RR}}^n)(\vec{x}) \in Y\right] = (1-q) \cdot (1-q)^{n-1}$$
$$\mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}_{\mathsf{RR}}^n)(\vec{x}') \in Y\right] = q \cdot (1-q)^{n-1}$$

We express one probability in terms of the other. In particular, we derive an equality involving a multiplicative term and an additive $\delta$ term:

$$\begin{aligned}
\mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}_{\mathsf{RR}}^n)(\vec{x}) \in Y\right] &= (1-2q) \cdot (1-q)^{n-1} + \mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}_{\mathsf{RR}}^n)(\vec{x}') \in Y\right] \\
&= \left((1-2q) \cdot (1-q)^{n-1} - \delta + \mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}_{\mathsf{RR}}^n)(\vec{x}') \in Y\right]\right) + \delta \\
&= \left(\frac{(1-2q) \cdot (1-q)^{n-1} - \delta}{\mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}_{\mathsf{RR}}^n)(\vec{x}') \in Y\right]} + 1\right) \cdot \mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}_{\mathsf{RR}}^n)(\vec{x}') \in Y\right] + \delta \\
&= \left(\frac{(1-2q) \cdot (1-q)^{n-1} - \delta}{q \cdot (1-q)^{n-1}} + 1\right) \cdot \mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}_{\mathsf{RR}}^n)(\vec{x}') \in Y\right] + \delta \qquad (4.6)
\end{aligned}$$

Taking $q = \frac{1}{2(e^\varepsilon+1)n} \ln \frac{1}{\delta}$, we will show that

$$(1-2q)\cdot(1-q)^{n-1} - \delta > (e^\varepsilon - 1)\cdot q \cdot (1-q)^{n-1}. \tag{4.7}$$

This will imply (4.6) is larger than $e^\varepsilon \cdot \mathbb{P}\Big[(\mathcal{S}\circ\mathcal{R}^n_{\mathsf{RR}})(\vec{x}') \in Y\Big] + \delta$; shuffling the messages of randomized response with parameter $q$ does not yield an $(\varepsilon,\delta)$-differentially private algorithm.

Notice that our upper bound on $\delta$ implies $\delta < \exp(-\frac{n}{2n-1})$ which can be rewritten as $\frac{\ln(1/\delta)}{n+\ln(1/\delta)} > \frac{1}{2n}$. Thus we have that

$$q < \frac{1}{e^\varepsilon+1}\cdot\frac{\ln(1/\delta)}{n+\ln(1/\delta)}$$

which can be rewritten as

$$\begin{aligned}\frac{1}{n}\ln\frac{1}{\delta} &> \frac{(e^\varepsilon+1)q}{1-(e^\varepsilon+1)q}\\ &\geq \ln\left(1+\frac{(e^\varepsilon+1)q}{1-(e^\varepsilon+1)q}\right)\\ &= \ln\left(\frac{1}{1-(e^\varepsilon+1)q}\right)\end{aligned}$$

Multiplying both sides by $n$ and raising $e$ to both sides gives us

$$\begin{aligned}\delta &< (1-(e^\varepsilon+1)q)^n\\ &< (1-(e^\varepsilon+1)q)(1-q)^{n-1}\end{aligned}$$

It is easy to verify that this is equivalent to (4.7). □

# Bibliography


[1] Jayadev Acharya, Clément L. Canonne, Cody Freitag, and Himanshu Tyagi. Test without trust: Optimal locally private distribution testing. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 2067–2076, 2019.

[2] Jayadev Acharya, Clément L. Canonne, Yanjun Han, Ziteng Sun, and Himanshu Tyagi. Domain compression and its application to randomness-optimal distributed goodness-of-fit. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 3–40. PMLR, 2020.

[3] Jayadev Acharya, Constantinos Daskalakis, and Gautam Kamath. Optimal testing for properties of distributions. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3591–3599, 2015.

[4] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Differentially private testing of identity and closeness of discrete distributions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 6879–6891, 2018.

[5] Gergely Ács and Claude Castelluccia. I have a dream! (differentially private smart metering). In Tomás Filler, Tomás Pevný, Scott Craver, and Andrew D. Ker, editors, *Information Hiding - 13th International Conference, IH 2011, Prague, Czech Republic, May 18-20, 2011, Revised Selected Papers*, volume 6958 of *Lecture Notes in Computer Science*, pages 118–132. Springer, 2011.

[6] Kareem Amin, Matthew Joseph, and Jieming Mao. Pan-private uniformity testing. *CoRR*, abs/1911.01452, 2019.

[7] Victor Balcer and Albert Cheu. Separating local & shuffled differential privacy via histograms. In Yael Tauman Kalai, Adam D. Smith, and Daniel Wichs, editors, *1st Conference on Information-Theoretic Cryptography, ITC 2020, June 17-19, 2020, Boston, MA, USA*, volume 163 of *LIPIcs*, pages 1:1–1:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[8] Victor Balcer, Albert Cheu, Matthew Joseph, and Jieming Mao. Connecting robust shuffle privacy and pan-privacy. *CoRR*, abs/2004.09481, 2020.

[9] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Differentially private summation with multi-message shuffling. *arXiv preprint arXiv:1906.09116*, 2019.

[10] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Improved summation from shuffling. *CoRR*, abs/1909.11225, 2019.

[11] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 638–667. Springer, 2019.

[12] Raef Bassily and Adam D. Smith. Local, private, efficient protocols for succinct histograms. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 127–135. ACM, 2015.

[13] Amos Beimel, Iftach Haitner, Kobbi Nissim, and Uri Stemmer. On the round complexity of the shuffle model. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 683–712. Springer, 2020.

[14] Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In David A. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 451–468. Springer, 2008.

[15] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*, pages 441–459. ACM, 2017.

[16] Jonas Böhler and Florian Kerschbaum. Secure multi-party computation of differentially private median. In Srdjan Capkun and Franziska Roesner, editors, *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, pages 2147–2164. USENIX Association, 2020.

[17] Mark Bun, Gautam Kamath, Thomas Steinke, and Zhiwei Steven Wu. Private hypothesis selection. In *Advances in Neural Information Processing Systems*, NeurIPS '19, pages 156–167, Vancouver, Canada, 2019.

[18] Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 369–380. ACM, 2016.

[19] Bryan Cai, Constantinos Daskalakis, and Gautam Kamath. Priv'it: Private and sample efficient identity testing. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 635–644, 2017.

[20] Clément L. Canonne. A short note on poisson tail bounds, 2017.

[21] TH Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In *European Symposium on Algorithms*, pages 277–288. Springer, 2012.

[22] Lijie Chen, Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. On distributed differential privacy and counting distinct elements. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 56:1–56:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[23] Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic*

*Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 375–403. Springer, 2019.

[24] Albert Cheu and Jonathan R. Ullman. The limits of pan privacy and shuffle privacy for learning and estimation. *CoRR*, abs/2009.08000, 2020.

[25] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In Frank Neven, Catriel Beeri, and Tova Milo, editors, *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, pages 202–210. ACM, 2003.

[26] John Duchi, Michael Jordan, and Martin Wainwright. Local privacy and statistical minimax rates. In *IEEE Symposium on Foundations of Computer Science*, FOCS '13, pages 429–438, Berkeley, CA, USA, 2013.

[27] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2006.

[28] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[29] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *Innovations in Computer Science (ICS)*, 2010.

[30] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 2014.

[31] Alexander Edmonds, Aleksandar Nikolov, and Jonathan Ullman. The power of factorization meisms in local and central differential privacy. In *ACM Symposium on the Theory of Computing*, STOC '20, pages 425–438, Chicago, IL, USA, 2020.

[32] Fabienne Eigner, Matteo Maffei, Ivan Pryvalov, Francesca Pampaloni, and Aniket Kate. Differentially private data aggregation with optimal utility. In Charles N. Payne Jr., Adam Hahn, Kevin R. B. Butler, and Micah Sherr, editors, *Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, December 8-12, 2014*, pages 316–325. ACM, 2014.

[33] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2468–2479. SIAM, 2019.

[34] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In Frank Neven, Catriel Beeri, and Tova Milo, editors, *PODS*, pages 211–222. ACM, 2003.

[35] Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. *CoRR*, abs/2012.12803, 2020.

[36] Simson Garfinkel, John M Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *acmqueue*, Nov 2018.

[37] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Pure differentially private summation from anonymous messages. *CoRR*, abs/2002.01919, 2020.

[38] Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. On the power of multiple anonymous messages. *IACR Cryptology ePrint Archive*, 2019:1382, 2019.

[39] Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Private aggregation from fewer anonymous messages. *CoRR*, abs/1909.11073, 2019.

[40] Badih Ghazi, Rasmus Pagh, and Ameya Velingker. Scalable and differentially private distributed aggregation in the shuffled model. *CoRR*, abs/1906.08320, 2019.

[41] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229. ACM, 1987.

[42] Slawomir Goryczka, Li Xiong, and Vaidy S. Sunderam. Secure multiparty aggregation with differential privacy: a comparative study. In Giovanna Guerrini, editor, *Joint 2013 EDBT/ICDT Conferences, EDBT/ICDT '13, Genoa, Italy, March 22, 2013, Workshop Proceedings*, pages 155–163. ACM, 2013.

[43] Moritz Hardt and Guy Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 61–70, Las vegas, NV, USA, 2014. IEEE Computer Society.

[44] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 239–248. IEEE, 2006.

[45] Matthew Joseph, Jieming Mao, Seth Neel, and Aaron Roth. The role of interactivity in local differential privacy. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 94–105. IEEE Computer Society, 2019.

[46] Matthew Joseph, Jieming Mao, and Aaron Roth. Exponential separations in local differential privacy. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 515–527. SIAM, 2020.

[47] Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. *CoRR*, abs/2102.06387, 2021.

[48] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International Conference on Machine Learning*, ICML '15, pages 1376–1385, Lille, France, 2015.

[49] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What can we learn privately? In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 531–540. IEEE Computer Society, 2008.

[50] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 94–103. IEEE Computer Society, 2007.

[51] Darakhshan J. Mir, S. Muthukrishnan, Aleksandar Nikolov, and Rebecca N. Wright. Pan-private algorithms via statistics on sketches. In *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece*, pages 37–48, 2011.

[52] Jack Murtagh and Salil Vadhan. The complexity of computing the optimal composition of differential privacy. In *Theory of Cryptography Conference*, TCC '16, pages 157–175, Beijing, China, 2016.

[53] Martin Pettai and Peeter Laud. Combining differential privacy and secure multiparty computation. In *Proceedings of the 31st Annual Computer Security Applications Conference, Los Angeles, CA, USA, December 7-11, 2015*, pages 421–430. ACM, 2015.

[54] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In Ahmed K. Elmagarmid and Divyakant Agrawal, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, pages 735–746. ACM, 2010.

[55] Elaine Shi, T.-H. Hubert Chan, Eleanor Gilbert Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6th February - 9th February 2011*. The Internet Society, 2011.

[56] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In Lorenzo Cavallaro, Johannes Kinder, Sadia Afroz, Battista Biggio, Nicholas Carlini, Yuval Elovici, and Asaf Shabtai, editors, *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2019, London, UK, November 15, 2019*, pages 1–11. ACM, 2019.

[57] Jonathan Ullman. Tight bounds for locally differentially private selection. *arXiv preprint arXiv:1802.02638*, 2018.

[58] Jonathan R. Ullman. Tight lower bounds for locally differentially private selection. *CoRR*, abs/1802.02638, 2018.

[59] Filipp Valovich and Francesco Aldà. Computational differential privacy from lattice-based cryptography. In Jerzy Kaczorowski, Josef Pieprzyk, and Jacek Pomykala, editors, *Number-Theoretic Methods in Cryptology - First International Conference, NuTMiC 2017, Warsaw, Poland, September 11-13, 2017, Revised Selected Papers*, volume 10737 of *Lecture Notes in Computer Science*, pages 121–141. Springer, 2017.

[60] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[61] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164. IEEE Computer Society, 1982.

[62] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167. IEEE Computer Society, 1986.

# Chapter A

# Appendix

## A.1 Miscellaneous Proofs

In this section, we provide proofs for some technical lemmas.

**Lemma A.1.1** (Restatement of Lemma 1.3.10)**.** *If local randomizer $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ is $(\varepsilon, \delta)$-differentially private, then there is a local randomizer $\mathcal{R}'$ that is $(2\varepsilon, 0)$-differentially private such that*

$$\forall x \in \mathcal{X} \;\; \mathrm{d}_{\mathrm{TV}}(\mathcal{R}(x), \mathcal{R}'(x)) \leq \delta$$

*Proof.* Fix an arbitrary element $\overline{x} \in \mathcal{X}$. We define $\mathcal{R}'(\overline{x})$ to have the same distribution as $\mathcal{R}(\overline{x})$.

For any other $x \in \mathcal{X}$, a lemma of Kairouz, Oh, and Viswanath [48][1] implies that there exists a tuple of distributions $(\tilde{\mathcal{R}}_0^{x,\overline{x}}, \tilde{\mathcal{R}}_1^{x,\overline{x}}, \tilde{\mathcal{R}}_\perp^{x,\overline{x}}, \tilde{\mathcal{R}}_\top^{x,\overline{x}})$ where

$$\mathcal{R}(x) = \left(\frac{e^\varepsilon(1-\delta)}{1+e^\varepsilon}\right)\tilde{\mathcal{R}}_0^{x,\overline{x}} + \left(\frac{1-\delta}{1+e^\varepsilon}\right)\tilde{\mathcal{R}}_1^{x,\overline{x}} + \delta\tilde{\mathcal{R}}_\perp^{x,\overline{x}}$$
$$\mathcal{R}(\overline{x}) = \left(\frac{1-\delta}{1+e^\varepsilon}\right)\tilde{\mathcal{R}}_0^{x,\overline{x}} + \left(\frac{e^\varepsilon(1-\delta)}{1+e^\varepsilon}\right)\tilde{\mathcal{R}}_1^{x,\overline{x}} + \delta\tilde{\mathcal{R}}_\top^{x,\overline{x}}$$

With this context, we define $\mathcal{R}'(x)$ to be the distribution

$$\mathcal{R}'(x) := \left(\frac{e^\varepsilon(1-\delta)}{1+e^\varepsilon}\right)\tilde{\mathcal{R}}_0^{x,\overline{x}} + \left(\frac{1-\delta}{1+e^\varepsilon}\right)\tilde{\mathcal{R}}_1^{x,\overline{x}} + \delta\tilde{\mathcal{R}}_\top^{x,\overline{x}}.$$

By construction, we have

$$\forall x \in \mathcal{X} \;\; \mathrm{d}_{\mathrm{TV}}(\mathcal{R}(x), \mathcal{R}'(x)) \leq \delta$$

Also by construction, we have

$$\forall R \subseteq \mathcal{R} \;\; e^{-\varepsilon} \leq \frac{\mathbb{P}[\mathcal{R}'(x) \in R]}{\mathbb{P}[\mathcal{R}'(\overline{x}) \in R]} \leq e^\varepsilon$$

which implies that, for every pair $x, x' \in \mathcal{X}$, we have

$$\forall R \subseteq \mathcal{R} \;\; \frac{\mathbb{P}[\mathcal{R}'(x) \in R]}{\mathbb{P}[\mathcal{R}'(x') \in R]} \leq e^{2\varepsilon},$$

as desired. □

[1] See also Murtagh and Vadhan [52, Lemma 3.2] for the precise form we use.

**Lemma A.1.2** (Restatement of Lemma 1.3.7)**.** *Let $f : \mathcal{X}^n \to \mathbb{Z}$ be a 1-sensitive function and fix any $\delta < 2e^{-9}$. For any $m \in \mathbb{N}$ and $p \in (0,1)$, let $\mathcal{M}_{f,m,p}$ denote the algorithm that samples $\eta \sim \mathbf{Bin}(m,p)$ and outputs $f(\vec{x}) + \eta$. If $m\min(p,1-p) > 13\ln\frac{2}{\delta}$, $\mathcal{M}_{f,m,p}$ is $(\varepsilon(m,p),\delta)$-differentially private, where*

$$\varepsilon(m,p) := \ln\left(1 + \sqrt{\frac{13\ln\frac{2}{\delta}}{m\min(p,1-p)}}\right) < \sqrt{\frac{13\ln\frac{2}{\delta}}{m\min(p,1-p)}}.$$

*Proof.* The bulk of this proof will argue that $\mathcal{M}_{f,m,p}$ is $(\overline{\varepsilon}(m,p),\delta)$-differentially private, where

$$\overline{\varepsilon}(m,p) := \ln\left(1 + \sqrt{\frac{13\ln\frac{2}{\delta}}{mp}}\right) < \sqrt{\frac{13\ln\frac{2}{\delta}}{mp}}.$$

This is sufficient to complete the proof whenever $p \le 1/2$. To handle the other case, we also will argue that privacy of $\mathcal{M}_{f,m,p}$ follows from privacy of $\mathcal{M}_{f,m,1-p}$.

Fix neighboring datasets $\vec{x} \sim \vec{x}'$. For any set $W \subset \mathbb{Z}$ and integer $z$, let $W - z$ denote the set $\{w - z \mid w \in W\}$. Using this notation, $W - f(\vec{x})$ contains exactly those values $\eta$ must take in order for $\mathcal{M}_{f,m,p}(\vec{x}) \in W$. Also note that $|W - f(\vec{x})| = |W - f(\vec{x}')|$. In fact, there is a bit $b \in \{\pm 1\}$ and a bijection between $W - f(\vec{x})$, $W - f(\vec{x}')$ such that $w' = w - b$ for each $w, w'$ in the bijection.

Define $u := \sqrt{3mp\ln\frac{2}{\delta}}$ and $I_u := [\lfloor mp - u \rfloor, \lceil mp + u \rceil]$. Because $mp > 3\ln(2/\delta)$, a Chernoff bound implies that $\mathop{\mathbb{P}}_{\eta\sim\mathbf{Bin}(m,p)}[\eta \notin I_u] < \delta$. We use this to decompose $\mathbb{P}\left[\mathcal{M}_{f,m,p}(\vec{x}) \in W\right]$:

$$\begin{aligned}\mathbb{P}\left[\mathcal{M}_{f,m,p}(\vec{x}) \in W\right] &= \mathbb{P}\left[\eta \in I_u \cap (W - f(\vec{x}))\right] + \mathbb{P}\left[\eta \in (W - f(\vec{x}))/I_u\right] \\ &\le \mathbb{P}\left[\eta \in I_u \cap (W - f(\vec{x}))\right] + \delta \end{aligned} \tag{A.1}$$

We define $I'_u := \{v - b \mid v \in I_u\}$. We rewrite the above using this interval:

$$\begin{aligned}(\text{A.1}) &= \left(\frac{\mathbb{P}\left[\eta \in I_u \cap (W - f(\vec{x}))\right]}{\mathbb{P}\left[\eta \in I'_u \cap (W - f(\vec{x}'))\right]}\right) \cdot \mathbb{P}\left[\eta \in I'_u \cap (W - f(\vec{x}'))\right] + \delta \\ &\le \left(\frac{\mathbb{P}\left[\eta \in I_u \cap (W - f(\vec{x}))\right]}{\mathbb{P}\left[\eta \in I'_u \cap (W - f(\vec{x}'))\right]}\right) \cdot \mathbb{P}\left[\mathcal{M}_{f,m,p}(\vec{x}') \in W\right] + \delta \\ &= \left(\frac{\sum_{r \in I_u \cap (W - f(\vec{x}))} \mathbb{P}\left[\eta = r\right]}{\sum_{r' \in I'_u \cap (W - f(\vec{x}'))} \mathbb{P}\left[\eta = r'\right]}\right) \cdot \mathbb{P}\left[\mathcal{M}_{f,m,p}(\vec{x}') \in W\right] + \delta\end{aligned}$$

Now it remains to show that the ratio in the above expression is at most $1 + \sqrt{\frac{13\ln\frac{2}{\delta}}{mp}}$.

Note that, by construction, there is a bijection between every $r \in I_u \cap (W - f(\vec{x}))$ and every $r' \in I'_u \cap (W - f(\vec{x}'))$ such that $r' = r - b$. As a consequence, it will suffice to bound the ratios $\frac{\mathbb{P}[\eta = r+1]}{\mathbb{P}[\eta = r]}$ and

$\frac{\mathbb{P}[\eta=r]}{\mathbb{P}[\eta=r+1]}$ for all $r \in [mp-u-1, mp+u]$. Notice that $mp > 13\ln\frac{2}{\delta}$ and $\delta < 2e^{-9}$ imply that $u > \sqrt{27\cdot 13}$.

$$
\begin{aligned}
&\frac{\mathbb{P}[\eta = r+1]}{\mathbb{P}[\eta = r]} \\
&= \frac{p}{1-p}\cdot\frac{m-r}{r+1} && (\eta \sim \mathbf{Bin}(m,p)) \\
&\le \frac{p}{1-p}\cdot\frac{m(1-p)+u+1}{mp-u} && (r \ge mp-u-1) \\
&= \frac{p}{1-p}\cdot\frac{u^2\cdot\frac{1-p}{3p\ln(2/\delta)}+u+1}{u^2/(3\ln\frac{2}{\delta})-u} \\
&= \frac{u^2\cdot\frac{1-p}{3\ln(2/\delta)}+(u+1)p}{u^2\cdot\frac{1-p}{3\ln(2/\delta)}-u(1-p)} = \frac{u(1-p)+3(1+\frac{1}{u})p\ln\frac{2}{\delta}}{u(1-p)-3(1-p)\ln\frac{2}{\delta}} \\
&= 1+\frac{3\ln\frac{2}{\delta}+\frac{3p}{u}\ln\frac{2}{\delta}}{u-3(1-p)\ln\frac{2}{\delta}} < 1+\frac{3(1+1/\sqrt{27\cdot 13})\ln\frac{2}{\delta}}{\sqrt{3mp\ln\frac{2}{\delta}}-3\ln\frac{2}{\delta}} && (u>\sqrt{27\cdot 13} \text{ and } p\in(0,1)) \\
&< 1+\frac{3(1+1/\sqrt{27\cdot 13})\ln\frac{2}{\delta}}{\sqrt{\frac{4}{5}mp\ln\frac{2}{\delta}}} < 1+\sqrt{\frac{13\ln\frac{2}{\delta}}{mp}}
\end{aligned}
$$

The inequality $\sqrt{3mp\ln\frac{2}{\delta}}-3\ln\frac{2}{\delta} > \sqrt{\frac{4}{5}mp\ln\frac{2}{\delta}}$ comes from $mp > 13\ln\frac{2}{\delta}$. We bound the second ratio using symmetric steps:

$$
\begin{aligned}
&\frac{\mathbb{P}[\eta = r]}{\mathbb{P}[\eta = r+1]} \\
&= \frac{1-p}{p}\cdot\frac{r+1}{m-r} && (\eta \sim \mathbf{Bin}(n,p)) \\
&\le \frac{1-p}{p}\cdot\frac{mp+u+1}{m(1-p)-u} && (r \le mp+u) \\
&= \frac{1-p}{p}\cdot\frac{u^2/(3\ln\frac{2}{\delta})+u+1}{u^2\cdot\frac{1-p}{3p\ln(2/\delta)}-u} \\
&= \frac{u^2\cdot\frac{1-p}{3\ln(2/\delta)}+(u+1)(1-p)}{u^2\cdot\frac{1-p}{3\ln(2/\delta)}-up} = \frac{u(1-p)+3(1+\frac{1}{u})(1-p)\ln\frac{2}{\delta}}{u(1-p)-3p\ln\frac{2}{\delta}} \\
&= 1+\frac{3\ln\frac{2}{\delta}+\frac{3(1-p)}{u}\ln\frac{2}{\delta}}{u-3p\ln\frac{2}{\delta}} < 1+\frac{3(1+1/\sqrt{27\cdot 13})\ln\frac{2}{\delta}}{\sqrt{3mp\ln\frac{2}{\delta}}-3\ln\frac{2}{\delta}} && (u>\sqrt{27\cdot 13} \text{ and } p\in(0,1)) \\
&< 1+\frac{3(1+1/\sqrt{27\cdot 13})\ln\frac{2}{\delta}}{\sqrt{\frac{4}{5}mp\ln\frac{2}{\delta}}} < 1+\sqrt{\frac{13\ln\frac{2}{\delta}}{mp}}
\end{aligned}
$$

Now we argue that privacy of $\mathcal{M}_{f,m,p}$ follows from privacy of $\mathcal{M}_{f,m,1-p}$. Fix any pair $\vec{x} \sim \vec{x}'$ and

$W \subseteq \mathbb{Z}$.

$$\begin{aligned}
\mathbb{P}\left[\mathcal{M}_{f,m,p}(\vec{x}) \in W\right] &= \mathop{\mathbb{P}}_{\eta\sim\mathbf{Bin}(m,p)}\left[\eta \in W - f(\vec{x})\right] \\
&= \mathop{\mathbb{P}}_{\eta\sim\mathbf{Bin}(m,1-p)}\left[m-\eta \in W - f(\vec{x})\right] && \text{(Symmetry)} \\
&= \mathop{\mathbb{P}}_{\eta\sim\mathbf{Bin}(m,1-p)}\left[\eta \in -W + (f(\vec{x})+m)\right] \\
&\le e^{\varepsilon}\cdot \mathop{\mathbb{P}}_{\eta\sim\mathbf{Bin}(m,1-p)}\left[\eta \in -W + (f(\vec{x}')+m)\right] + \delta \\
&= e^{\varepsilon}\cdot \mathop{\mathbb{P}}_{\eta\sim\mathbf{Bin}(m,1-p)}\left[m-\eta \in W - f(\vec{x}')\right] + \delta \\
&= e^{\varepsilon}\cdot \mathbb{P}\left[\mathcal{M}_{f,m,p}(\vec{x}') \in W\right] + \delta && \square
\end{aligned}$$

## A.2 Equating the Shuffle and Secure Aggregation Models

We state the equivalence as two lemmas that losslessly translate between the models.

**Lemma A.2.1.** *Fix any finite set $\mathcal{Y} \neq \emptyset$ and let $d = |\mathcal{Y}|$. For any shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ using message space $\mathcal{Y}$, there is a secure aggregation protocol $\hat{\mathcal{P}} = (\hat{\mathcal{R}}, \hat{\mathcal{S}}_{d,\ge 0}, \hat{\mathcal{A}})$ such that*

- *$\mathcal{P}$ is $(\tilde{\varepsilon}, \tilde{\delta}, \tau)$-robustly private for $n$ users if and only if $\hat{\mathcal{P}}$ is $(\tilde{\varepsilon}, \tilde{\delta}, \tau)$-robustly private for $n$ users.*
- *On any input $\vec{x} \in \mathcal{X}^n$, $\hat{\mathcal{P}}(\vec{x})$ is identically distributed with $\mathcal{P}(\vec{x})$.*

**Lemma A.2.2.** *Fix any finite set $\mathcal{Y} \neq \emptyset$ and let $d = |\mathcal{Y}|$. For any secure aggregation protocol $\hat{\mathcal{P}} = (\hat{\mathcal{R}}, \hat{\mathcal{S}}_{d,\ge 0}, \hat{\mathcal{A}})$, there is a shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ using message space $\mathcal{Y}$ such that*

- *$\hat{\mathcal{P}}$ is $(\tilde{\varepsilon}, \tilde{\delta}, \tau)$-robustly private for $n$ users if and only if $\mathcal{P}$ is $(\tilde{\varepsilon}, \tilde{\delta}, \tau)$-robustly private for $n$ users.*
- *On any input $\vec{x} \in \mathcal{X}^n$, $\mathcal{P}(\vec{x})$ is identically distributed with $\hat{\mathcal{P}}(\vec{x})$.*

*Proof of Lemma A.2.1.* We assume some bijection between $y \in \mathcal{Y}$ and $t[y] \in [d]$. Let $\hat{\mathcal{R}}(x)$ be the algorithm that executes $\mathcal{R}(x)$ and reports the histogram of message values. That is, if the message $y$ occurs 5 times, the $t[y]$-th location in the message vector is the integer 5.

Let $F : \mathcal{Y}^* \to \mathbb{Z}^d_{\ge 0}$ be the algorithm which, on input $\vec{y}$, reports the histogram of the messages $\vec{y}$. On any input $\vec{x}$, we will argue that $(F \circ \mathcal{S} \circ \mathcal{R}^{\gamma n})(\vec{x})$ is identically distributed with $(\hat{\mathcal{S}}_{d,\ge 0} \circ \hat{\mathcal{R}}^{\gamma n})(\vec{x})$. So when $\mathcal{P}$ is robustly shuffle private, $\hat{\mathcal{P}}$ is also robustly shuffle private by post-processing (Fact 1.3.5).

For any vector $\vec{h} \in \mathbb{Z}^d_{\ge 0}$,

$$\begin{aligned}
&\mathbb{P}\left[(F \circ \mathcal{S} \circ \mathcal{R}^{\gamma n})(\vec{x}) = \vec{h}\right] \\
&= \mathbb{P}\left[(F \circ \mathcal{R}^{\gamma n})(\vec{x}) = \vec{h}\right] && (\mathcal{S} \text{ only permutes}) \\
&= \mathbb{P}\left[(\hat{\mathcal{S}}_{d,\ge 0} \circ \hat{\mathcal{R}}^{\gamma n})(\vec{x}) = \vec{h}\right] && (\text{Defn. of } F, \hat{\mathcal{S}})
\end{aligned}$$

Let $\hat{F} : \mathbb{Z}^d_{\ge 0} \to \mathcal{Y}^*$ be the algorithm which, on input $\vec{h}$, samples a random permutation of messages $\vec{y}$ such that the frequency of any message $y$ is $h_{t[y]}$. On any input $\vec{x}$, we will argue that $(\hat{F} \circ \hat{\mathcal{S}}_{d,\ge 0} \circ \hat{\mathcal{R}}^{\gamma n})(\vec{x})$ is identically distributed with $(\mathcal{S} \circ \mathcal{R}^{\gamma n})(\vec{x})$. So when $\hat{\mathcal{P}}$ is robustly shuffle private, $\mathcal{P}$ is also robustly shuffle private, again by post-processing.

Let $p(\vec{y}, m)$ be the function that reports all possible ways of placing the balls $y_1, y_2, \ldots$ into $m$ bins. That is, $\vec{p} \in p(\vec{y}, m)$ iff $p_i$ is a subsequence of $\vec{y}$, all $p_i$ are disjoint, and $\vec{y}$ can be formed by concatenating and

permuting $p_1, \ldots, p_m$. For any vector $\vec{y} \in \mathcal{Y}^*$,

$$\begin{aligned}
&\mathbb{P}\left[(\hat{F} \circ \hat{\mathcal{S}} \circ \hat{\mathcal{R}}^{\gamma n})(\vec{x}) = \vec{y}\right] \\
&= \mathbb{P}\left[\hat{F}(F(\vec{y})) = \vec{y}\right] \cdot \mathbb{P}\left[(\hat{\mathcal{S}} \circ \hat{\mathcal{R}}^{\gamma n})(\vec{x}) = F(\vec{y})\right] \\
&= \frac{1}{|\vec{y}|!} \cdot \sum_{\vec{p} \in p(|\vec{y}|, \gamma n)} \prod_{i=1}^{\gamma n} \mathbb{P}\left[\mathcal{R}(x_i) = p_i\right] \\
&= \mathbb{P}\left[(\mathcal{S} \circ \mathcal{R}^{\gamma n})(\vec{x}) = \vec{y}\right]
\end{aligned}$$

If we define the analyzer $\hat{\mathcal{A}}$ such that $\hat{\mathcal{A}}(\vec{h}) := \mathcal{A}(\hat{F}(\vec{h}))$, then $(\hat{\mathcal{A}} \circ \hat{\mathcal{S}} \circ \hat{\mathcal{R}}^n) = (\mathcal{A} \circ \mathcal{S} \circ \mathcal{R}^n)$ which completes the proof. □

*Proof of Lemma A.2.2.* The proof is symmetric to the preceding one. Let $\mathcal{R}(x)$ be the algorithm that executes $\hat{\mathcal{R}}(x)$ and then outputs a uniformly random vector of messages whose frequencies obey the histogram. That is, if the $t[y]$-th coordinate is 5, then the randomizer reports the message $y$ 5 times. We use the same algorithms $F, \hat{F}$ and define $\mathcal{A}(\vec{y}) := \hat{\mathcal{A}}(F(\vec{y}))$. □

## A.3 Proofs for Distinct Elements Protocol $\mathcal{P}_{\mathsf{DE}}$

**Lemma** (Restatement of Lemma 2.4.6)**.** Fix any integers $m \le n$ and real number $p^* \in [0, 1/2]$. If we assign $p \leftarrow \frac{1-(1-2p^*)^{1/n}}{2}$ and sample i.i.d. $X_1, \ldots, X_m \sim \mathbf{Ber}(p)$, then the sum $X := \sum_{i=1}^m X_i \pmod 2$ is distributed as

$$\mathbf{Ber}\left(\frac{1-(1-2p^*)^{m/n}}{2}\right).$$

*Proof.* For all $i \in [m]$, define $Y_i := 1 - 2X_i \in \{\pm 1\}$, and define $Y := \prod_{i=1}^m Y_i$. Then

$$\mathbb{P}[X = 1] = \mathbb{P}[Y = -1] = \frac{1 - \mathbb{E}[Y]}{2}.$$

Now, by independence, we rewrite

$$\mathbb{E}[Y] = \prod_{i=1}^m \mathbb{E}[Y_i] = \prod_{i=1}^m (1 - 2\mathbb{E}[X_i]) = (1-2p)^m$$

Thus, we have that

$$\begin{aligned}
\mathbb{P}[X = 1] &= \frac{1-(1-2p)^m}{2} \\
&= \frac{1-(1-(1-(1-2p^*)^{1/n}))^m}{2} \\
&= \frac{1-(1-2p^*)^{m/n}}{2}
\end{aligned}$$

□

**Lemma A.3.1.** *For any $\varepsilon > 0$ and $\gamma \in (0, 1]$,*

$$\ln\left(\frac{1}{1-(1-e^{-\varepsilon})^\gamma}\right) \le \varepsilon + \ln(1/\gamma)$$

*and when $\varepsilon \in (0, 1)$,*

$$\ln\left(\frac{1}{1-(1-e^{-\varepsilon})^\gamma}\right) \le \frac{\varepsilon^\gamma}{\gamma}.$$

*Proof.* By Bernoulli's inequality, $(1-e^{-\varepsilon})^\gamma \le 1-\gamma e^{-\varepsilon}$ which implies that

$$\begin{aligned}\ln\left(\frac{1}{1-(1-e^{-\varepsilon})^\gamma}\right) &\le \ln\left(\frac{1}{1-(1-\gamma e^{-\varepsilon})}\right)\\ &= \varepsilon + \ln(1/\gamma)\end{aligned}$$

We move on to the regime where $\varepsilon \in (0,1)$. Here, we will prove that $1-e^{-\varepsilon} \le \left(1-e^{-\varepsilon\gamma/\gamma}\right)^{1/\gamma}$ from which the desired bound follows by substitution. To prove this observation, we first apply the substitution $\mu = 1/\gamma \in [1,\infty)$:

$$1-e^{-\varepsilon} \le \left(1-e^{-\mu\varepsilon^{1/\mu}}\right)^\mu. \tag{A.2}$$

Notice that equality holds at $\mu = 1$ for all $\varepsilon \in (0,1]$. Let $f(\varepsilon,\mu)$ be the expression on the RHS of Equation A.2. To prove our observation, it suffices to show that $f$ is non-decreasing in $\mu$ for every $\varepsilon \in (0,1]$. We take the first derivative:

$$\begin{aligned}&\frac{\partial f(\varepsilon,\mu)}{\partial \mu}\\ &= f(\varepsilon,\mu)\cdot\frac{\partial}{\partial\mu}\left(\mu\cdot\ln\left(1-e^{-\mu\varepsilon^{1/\mu}}\right)\right)\\ &= f(\varepsilon,\mu)\cdot\left(\ln\left(1-e^{-\mu\varepsilon^{1/\mu}}\right)+\frac{\varepsilon^{1/\mu}\cdot(\mu+\ln(1/\varepsilon))}{e^{\mu\varepsilon^{1/\mu}}-1}\right).\end{aligned}$$

Let $g(\varepsilon,\mu) := \left(e^{\mu\varepsilon^{1/\mu}}-1\right)\cdot\ln\left(1-e^{-\mu\varepsilon^{1/\mu}}\right)+\varepsilon^{1/\mu}\cdot(\mu+\ln(1/\varepsilon))$. Then

$$\frac{\partial f(\varepsilon,\mu)}{\partial\mu} = \frac{f(\varepsilon,\mu)}{e^{\mu\varepsilon^{1/\mu}}-1}\cdot g(\varepsilon,\mu).$$

To show that $\frac{\partial}{\partial\mu}f(\varepsilon,\mu)\ge 0$, it suffices to show that $g(\varepsilon,\mu)\ge 0$ for all $\varepsilon\in(0,1]$ and $\mu\ge 1$ since $f(\varepsilon,\mu)>0$ and $e^{\mu\varepsilon^{1/\mu}}-1>0$ for all $\varepsilon\in(0,1]$ and $\mu\ge 1$. To show that $g(\varepsilon,\mu)\ge 0$, we show that $g(\varepsilon,\mu)\ge 0$ near the boundary of the domain (i.e. as $\varepsilon\to 0^+$ and at $\varepsilon=1$) for every $\mu\ge 1$ and that $g(\varepsilon,\mu)$ approaches this boundary from the proper direction as a function of $\varepsilon$. Now,

$$\frac{\partial g(\varepsilon,\mu)}{\partial\varepsilon} = \varepsilon^{(1/\mu)-1}\cdot\left(e^{\mu\varepsilon^{1/\mu}}\ln\left(1-e^{-\mu\varepsilon^{1/\mu}}\right)+\frac{\ln(1/\varepsilon)}{\mu}+1\right).$$

Let $h(\varepsilon,\mu) := \mu e^{\mu\varepsilon^{1/\mu}}\ln\left(1-e^{-\mu\varepsilon^{1/\mu}}\right)+\ln(1/\varepsilon)+\mu$. Then

$$\frac{\partial g(\varepsilon,\mu)}{\partial\varepsilon} = \frac{\varepsilon^{(1/\mu)-1}}{\mu}\cdot h(\varepsilon,\mu)$$

and

$$\begin{aligned}\frac{\partial h(\varepsilon,\mu)}{\partial\varepsilon} &= \mu e^{\mu\varepsilon^{1/\mu}}\varepsilon^{(1/\mu)-1}\left(\ln\left(1-e^{-\mu\varepsilon^{1/\mu}}\right)+\frac{1}{e^{\mu\varepsilon^{1/\mu}}-1}\right)-\frac{1}{\varepsilon}\\ &\le \frac{\mu\varepsilon^{(1/\mu)-1}}{e^{\mu\varepsilon^{1/\mu}}-1}-\frac{1}{\varepsilon} && (\ln(1+x)\le x \text{ for } x>-1)\\ &\le 0. && (e^x-1\ge x \text{ for } x\in\mathbb{R})\end{aligned}$$

Thus, $h$ is non-increasing in $\varepsilon$. Now, for all $\varepsilon'\in(0,1]$ and $\mu\ge 1$, we perform case analysis:

- If $h(\varepsilon', \mu) \geq 0$, then $\frac{\partial}{\partial \varepsilon} g(\varepsilon^*, \mu) \geq 0$ at every $\varepsilon^* \in (0, \varepsilon']$ which implies $g$ is non-decreasing in $\varepsilon$ on $(0, \varepsilon']$. Thus,

$$\begin{aligned} g(\varepsilon', \mu) &\geq \lim_{\varepsilon \to 0^+} g(\varepsilon, \mu) \\ &= \lim_{\varepsilon \to 0^+} \frac{\ln\left(1 - e^{-\mu\varepsilon^{1/\mu}}\right)}{\left(e^{\mu\varepsilon^{1/\mu}} - 1\right)^{-1}} + \lim_{\varepsilon \to 0^+} \varepsilon^{1/\mu}\mu + \lim_{\varepsilon \to 0^+} \frac{\ln(1/\varepsilon)}{\varepsilon^{-1/\mu}} \\ &= \lim_{\varepsilon \to 0^+} \frac{\varepsilon^{(1/\mu)-1}\left(e^{\mu\varepsilon^{1/\mu}} - 1\right)^{-1}}{-\varepsilon^{(1/\mu)-1} \cdot e^{\mu\varepsilon^{1/\mu}}\left(e^{\mu\varepsilon^{1/\mu}} - 1\right)^{-2}} + \lim_{\varepsilon \to 0^+} \frac{-1/\varepsilon}{-\varepsilon^{-1/\mu}/(\mu\varepsilon)} && \text{(L'Hôpital's rule)} \\ &= 0. \end{aligned}$$

- If $h(\varepsilon', \mu) < 0$, then $\frac{\partial}{\partial \varepsilon} g(\varepsilon^*, \mu) \leq 0$ at every $\varepsilon^* \in [\varepsilon', 1]$ which implies $g$ is non-increasing in $\varepsilon$ on $[\varepsilon', 1]$. Thus,

$$\begin{aligned} g(\varepsilon', \mu) &\geq g(1, \mu) \\ &= (e^{\mu} - 1) \cdot \ln(1 - e^{-\mu}) + \mu \\ &\geq -1 + \mu && (\ln(1+x) \geq \tfrac{x}{x+1} \text{ for } x > -1) \\ &\geq 0. \end{aligned}$$

Thus, $g(\varepsilon, \mu) \geq 0$ on the desired domain which concludes the proof. □

## A.4 Proofs for Uniformity Testing Protocol $\mathcal{P}_{\mathsf{UT}}$

Here, we provide proofs for the technical claims made in the proof of Theorem 2.5.3.

**Claim A.4.1** (Restatement of Claim 2.5.4)**.** *Sample $n \sim \mathbf{Pois}(m)$ and $\vec{x} \sim \mathbf{U}^n$. There is a constant $\kappa$ such that when $m > \kappa d^{1/2}/\alpha^2$, the following inequalities hold in an execution of $\mathcal{P}_{\mathsf{UT}}(\vec{x})$:*

$$\mathbb{P}\left[Z > \frac{3\alpha^2 m}{250}\right] < 1/40$$

$$\mathbb{P}\left[A > \frac{d^2\lambda}{4m} + \sqrt{\frac{20d^3\lambda^2}{m^2}}\right] < 1/40$$

$$\mathbb{P}\left[C < -\sqrt{\frac{10d^3\lambda}{m^2}}\right] < 1/40$$

*Proof.* Recall $\eta_j$ is the noise introduced by $\mathcal{P}_{\mathsf{SYM}}$ to the count $c_j(\vec{x})$. From Claim 2.1.7, the first four moments are $0, \lambda/4, 0, 3\lambda^2/10 + 7\lambda/40$.

The expectation of $A$ immediately follows from linearity and the second moment of $\eta_j$:

$$\mathbb{E}[A] = \frac{d}{m}\sum_{j=1}^{d} \mathbb{E}\left[\eta_j^2\right] = \frac{d^2\lambda}{4m}$$

We derive the expectation of $B$ in a similar way:

$$\mathbb{E}[C] = \frac{d}{m}\sum_{j=1}^{d} \mathbb{E}\left[\eta_j\right] = 0$$

The variance calculations follow essentially the same recipe:

$$\begin{aligned}
\mathrm{Var}[A] &= \frac{d^2}{m^2}\sum_{j=1}^{d}\mathrm{Var}\left[\eta_j^2\right] && \text{(Independence)}\\
&= \frac{d^2}{m^2}\sum_{j=1}^{d}\left(\mathbb{E}\left[\eta_j^4\right]-\mathbb{E}\left[\eta_j^2\right]^2\right)\\
&= \frac{d^3}{m^2}\left(\frac{3\lambda^2}{10}+\frac{7\lambda}{40}-\frac{\lambda^2}{16}\right)\\
&\le \frac{d^3\lambda^2}{2m^2} && (\lambda>1)\\
\mathrm{Var}[C] &= \frac{d^2}{m^2}\sum_{j=1}^{d}\mathrm{Var}\left[\eta_j\right] && \text{(Independence)}\\
&= \frac{d^3\lambda}{4m^2}
\end{aligned}$$

As observed in [6], the analysis by Acharya Daskalakis and Kamath [3] implies

$$\begin{aligned}
\mathbb{E}[Z] &\le \frac{\alpha^2 m}{500}\\
\mathrm{Var}[Z] &\le \frac{\alpha^4 m^2}{500000}
\end{aligned}$$

Chebyshev's inequality completes the proof. □

**Claim A.4.2** (Restatement of 2.5.6)**.** *Sample $n\sim\mathbf{Pois}(m)$ and $\vec{x}\sim\mathbf{D}^n$ where $\|\mathbf{D}-\mathbf{U}\|_{\mathrm{TV}}>\alpha$. There is a constant $c$ such that when $m>cd^{1/2}/\alpha^2$, the following inequalities hold in an execution of $\mathcal{P}_{\mathsf{UT}}(\vec{x})$:*

$$\begin{aligned}
\mathbb{P}\left[Z<\frac{\alpha^2 m}{15}\right] &< 1/30\\
\mathbb{P}\left[A<\frac{d^2\lambda}{4m}-\sqrt{\frac{15d^3\lambda^2}{m^2}}\right] &< 1/30\\
\mathbb{P}\left[C>\sqrt{\frac{15d^3\lambda}{2m^2}}\right] &< 1/30
\end{aligned}$$

*Proof.* As with the prior proof, we use the analysis of [3] to derive

$$\mathbb{E}[Z]\ge\frac{\alpha^2 m}{5} \tag{A.3}$$

$$\mathrm{Var}[Z]\le\frac{\mathbb{E}[Z]^2}{100} \tag{A.4}$$

By Chebyshev's inequality, we have that the following holds with probability $\ge 29/30$:

$$\begin{aligned}
Z &\ge \mathbb{E}[Z]-\sqrt{\frac{\mathbb{E}[Z]^2}{100}\cdot 30}\\
&= (1-\sqrt{3/10})\mathbb{E}[Z]\\
&\ge (1-\sqrt{3/10})\frac{\alpha^2 m}{5}\\
&> \frac{\alpha^2 m}{15}
\end{aligned}$$

Note that the expressions for $\mathbb{E}[A], \mathbb{E}[C], \mathrm{Var}[A], \mathrm{Var}[C]$ we obtained in the previous claim are true regardless of the identity of $\mathbf{D}$. Thus, they hold here as well. Chebyshev's inequality again completes the proof. □

**Claim A.4.3** (Restatement of Claim 2.5.5)**.** *Let $\eta_1, \ldots, \eta_d$ be independent random variables where each $\eta_j$ is symmetrically distributed over the set $\{\ldots, -3/2, -1, -1/2, 0, 1/2, 1, 3/2, \ldots\}$ with mean zero. For any coefficients $a_1, \ldots, a_d \in \mathbb{R}$, the random variable $\sum_{j=1}^{d} \eta_j \cdot a_j$ is symmetrically distributed with mean zero.*

*Proof.* By linearity of expectation, the mean of $\sum_{j=1}^{d} \eta_j \cdot a_j$ is zero. Next we argue that the distribution of each term $\eta_j \cdot a_j$ is symmetric: for any $v \in \mathbb{R}$,

$$\begin{aligned}
\mathbb{P}\left[\eta_j \cdot a_j = v\right] &= \mathbb{P}\left[\eta_j = v/a_j\right] \\
&= \mathbb{P}\left[\eta_j = -v/a_j\right] && \text{(Symmetry)} \\
&= \mathbb{P}\left[\eta_j \cdot a_j = -v\right]
\end{aligned}$$

Now it will suffice to prove the following: if $\mathbf{D}, \mathbf{D}'$ are symmetric distributions over countable supports $T, T'$, their convolution is symmetric. For any $v \in \mathbb{R}$,

$$\begin{aligned}
\mathbb{P}_{t \sim \mathbf{D}, t' \sim \mathbf{D}'}\left[t + t' = v\right] &= \sum_{u \in T} \mathbb{P}\left[t = u\right] \cdot \mathbb{P}\left[t' = v - u\right] \\
&= \sum_{u \in T} \mathbb{P}\left[t = -u\right] \cdot \mathbb{P}\left[t' = -(v - u)\right] \\
&= \mathbb{P}\left[t + t' = -v\right]
\end{aligned}$$

□

## A.5 Deferred Lower Bound Proofs

In this section, we prove the lower bounds for hypothesis testing, sparse mean estimation, and parity release.

### A.5.1 Simple Hypothesis Testing

**Theorem A.5.1** (Restatement of Theorem 3.3.14)**.** *If $\mathcal{Q}$ is an $(\varepsilon, \delta)$-internally private algorithm that solves $d$-wise simple hypothesis testing with error $\alpha$ and $\delta \log d/\delta \ll \alpha^2 \varepsilon^2 / d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.*

*Proof.* Consider the set of distributions $\{\mathbf{U}\} \cup \mathcal{D}_{d,1,\alpha}$. Note that this is a family of $2d+1$ distributions. From Fact 3.3.8, its size is $2d+1$. Later in this section, we will prove the following:

**Claim A.5.2.** *For any $\mathbf{D} \neq \mathbf{D}' \in \{\mathbf{U}\} \cup \mathcal{D}_{d,1,\alpha}$, $\mathrm{d_{TV}}(\mathbf{D}, \mathbf{D}') \geq \alpha$.*

The upshot is that $\{\mathbf{U}\} \cup \mathcal{D}_{d,1,\alpha}$ is a valid set of distributions for $(2d+1)$-wise hypothesis testing. We now argue that the accuracy of $M$ for this problem instance implies that we can invoke Theorem 3.3.11.

To do so, let $\mathbf{D}_{d,L,B,\alpha}$ denote a distribution chosen uniformly at random from $\mathcal{D}_{d,1,\alpha}$. We show that the

total variation distance between $\mathcal{Q}(\mathbf{U}^n)$ and $\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})$ is at least some positive constant.

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}(\mathbf{U}^n), \mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})) \\
&= \max_{\mathcal{D} \subseteq \{\mathbf{U}\} \cup \mathcal{D}_{d,k,\alpha}} \left| \mathbb{P}\left[\mathcal{Q}(\mathbf{U}^n) \in \mathcal{D}\right] - \mathbb{P}\left[\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha}) \in \mathcal{D}\right] \right| \\
&\geq \mathbb{P}\left[\mathcal{Q}(\mathbf{U}^n) \in \{\mathbf{U}\}\right] - \mathbb{P}\left[\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha}) \in \{\mathbf{U}\}\right] \\
&\geq \mathbb{P}\left[\mathcal{Q}(\mathbf{U}^n) \in \{\mathbf{U}\}\right] - \frac{1}{100} \\
&\geq \frac{99}{100} - \frac{1}{100} = \frac{49}{50}
\end{aligned}$$

To obtain the second inequality, we first observe that $\mathbf{D}_{d,t,b,\alpha} \neq \mathbf{U}$ for every $t, b$ so $\mathbf{U}$ would be an incorrect output. Then we use the fact that $\mathcal{Q}$ solves simple hypothesis testing: it is incorrect with probability at most $1/100$. The same reasoning yields the third inequality.

From Theorem 3.3.11, we conclude that $n = \Omega\left(\frac{1}{\varepsilon \|\mathcal{D}_{d,1,\alpha}\|_{\infty \to 2}}\right) = \Omega\left(\sqrt{d}/\alpha\varepsilon\right)$. This lower bound holds for a family of $2d+1$ distributions, so the claimed result follows by rescaling $d$. □

The next theorem adapts our proof to the robust shuffle privacy setting:

**Theorem A.5.3** (Restatement of Theorem 3.3.15)**.** *If $\mathcal{D}$ is an $(\varepsilon, \delta)$-robustly shuffle private protocol that solves $d$-wise simple hypothesis testing with error $\alpha$ and $\delta \log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.*

*Proof.* As before, let $\mathbf{D}_{d,L,B,\alpha}$ denote a distribution chosen uniformly at random from $\mathcal{D}_{d,1,\alpha}$. Let $\Pi$ denote an algorithm in the shuffle model that solves $(2d+1)$-wise simple hypothesis testing with accuracy $2\alpha/9$.

Let $\mathcal{Q}_{\mathcal{D}}$ denote the $(\varepsilon, \delta)$-internally private algorithm guaranteed by Lemma 3.2.2. We will lower bound the total variation distance between $\mathcal{Q}_{\mathcal{D}}(\mathbf{U}^{n/3})$ and $\mathcal{Q}_{\mathcal{D}}(\mathbf{D}^{n/3}_{d,L,B,\alpha})$.

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}_{\mathcal{D}}(\mathbf{U}^{n/3}), \mathcal{Q}_{\mathcal{D}}(\mathbf{D}^{n/3}_{d,L,B,\alpha})) \\
&\geq \mathbb{P}\left[\mathcal{Q}_{\mathcal{D}}(\mathbf{U}^{n/3}) \in \{\mathbf{U}\}\right] - \mathbb{P}\left[\mathcal{Q}_{\mathcal{D}}(\mathbf{D}^{n/3}_{d,L,B,\alpha}) \in \{\mathbf{U}\}\right] \\
&\geq \mathbb{P}\left[\Pi(\mathbf{U}^n) \in \{\mathbf{U}\}\right] - \mathbb{P}\left[\Pi(\mathbf{D}^n_{d,L,B,2\alpha/9}) \in \{\mathbf{U}\}\right] - \frac{1}{6} && \text{(Lemma 3.2.2)} \\
&\geq \frac{49}{50} - \frac{1}{6} = \frac{61}{75}
\end{aligned}$$

The third inequality comes from repeating the analysis in the proof of Theorem 3.3.14. Since $\mathcal{Q}_{\mathcal{D}}$ is an $(\varepsilon, \delta)$-internally private algorithm such that

$$\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}_{\mathcal{D}}(\mathbf{U}^{n/3}), \mathcal{Q}_{\mathcal{D}}(\mathbf{D}^{n/3}_{d,L,B,\alpha}))$$

is at least a positive constant, we invoke Theorem 3.3.11 to conclude that $n = \Omega(\sqrt{d}/\alpha\varepsilon)$. The claimed theorem follows by rescaling $\alpha$ and $d$. □

*Proof of Claim A.5.2.* We first compute the distance between the uniform distribution and $\mathbf{D}_{d,\{j\},b,\alpha}$ (for

generic $j \in [d]$ and $b \in \{\pm 1\}$):

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathbf{U}, \mathbf{D}_{d,\{j\},b,\alpha}) \\
&= \frac{1}{2}\|\mathbf{U} - \mathbf{D}_{d,\{j\},b,\alpha}\|1 \\
&= \frac{1}{2}\left(\sum_{x\in\mathcal{X}, x_j = b} |2^{-d} - (1+2\alpha)2^{-d}| + \sum_{x\in\mathcal{X}, x_j = -b} |2^{-d} - (1-2\alpha)2^{-d}|\right) \\
&= \frac{1}{2}\left(\alpha \cdot 2^{-d+1} \cdot 2^{d-1} + \alpha \cdot 2^{-d+1} \cdot 2^{d-1}\right) \\
&= \alpha
\end{aligned}$$

For any $j, j' \in [d]$ and any $b, b' \in \{\pm 1\}$, we calculate the distance $\mathrm{d}_{\mathrm{TV}}(\mathbf{D}_{d,\{j\},b,\alpha}, \mathbf{D}_{d,\{j'\},b',\alpha})$ via case analysis. When $j \neq j'$,

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathbf{D}_{d,\{j\},b,\alpha}, \mathbf{D}_{d,\{j'\},b',\alpha}) \\
&= \frac{1}{2}\|\mathbf{D}_{d,\{j\},b,\alpha} - \mathbf{D}_{d,\{j'\},b',\alpha}\|1 \\
&= \frac{1}{2}\cdot \sum_{\substack{x_j = b \\ x_{j'} = b'}} |(1+2\alpha)2^{-d} - (1+2\alpha)2^{-d}| + \frac{1}{2}\cdot \sum_{\substack{x_j \neq b \\ x_{j'} \neq b'}} |(1-2\alpha)2^{-d} - (1-2\alpha)2^{-d}| \\
&\quad + \frac{1}{2}\cdot \sum_{\substack{x_j = b \\ x_{j'} \neq b'}} |(1+2\alpha)2^{-d} - (1-2\alpha)2^{-d}| + \frac{1}{2}\cdot \sum_{\substack{x_j \neq b \\ x_{j'} = b'}} |(1-2\alpha)2^{-d} - (1+2\alpha)2^{-d}| \\
&= \frac{1}{2}\cdot \sum_{\substack{x_j = b \\ x_{j'} \neq b'}} \alpha \cdot 2^{-d+2} + \frac{1}{2}\cdot \sum_{\substack{x_j \neq b \\ x_{j'} = b'}} \alpha \cdot 2^{-d+2} \\
&= \frac{1}{2}\left(\alpha \cdot 2^{-d+2} \cdot 2^{d-2} + \alpha \cdot 2^{-d+2} \cdot 2^{d-2}\right) \\
&= \alpha
\end{aligned}$$

When $j = j'$ but $b \neq b'$, we take $b = +1$ and $b' = -1$ without loss of generality.

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathbf{D}_{d,\{j\},+1,\alpha}, \mathbf{D}_{d,\{j'\},-1,\alpha}) \\
&= \frac{1}{2}\|\mathbf{D}_{d,\{j\},+1,\alpha} - \mathbf{D}_{d,\{j'\},-1,\alpha}\|1 \\
&= \frac{1}{2}\left(\sum_{x_j = +1} |(1+2\alpha)2^{-d} - (1-2\alpha)2^{-d}| + \sum_{x_j = -1} |(1-2\alpha)2^{-d} - (1+2\alpha)2^{-d}|\right) \\
&= \frac{1}{2}\left(\alpha \cdot 2^{-d+2} \cdot 2^{d-1} + \alpha \cdot 2^{-d+2} \cdot 2^{d-1}\right) \\
&= 2\alpha
\end{aligned}$$

□

## A.5.2 Sparse Mean Estimation

**Theorem A.5.4** (Restatement of Theorem 3.3.17)**.** *If $\mathcal{Q}$ is an $(\varepsilon, \delta)$-internally private algorithm that solves $(d, 1, \alpha)$-sparse mean estimation and $\delta \log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.*

*Proof.* As before, let $\mathbf{D}_{d,L,B,\alpha}$ denote a distribution chosen uniformly at random from $\mathcal{D}_{d,1,\alpha}$. By construction, the mean of this distribution is 1-sparse, namely it is $B \cdot \vec{e}_L$ wehre $\vec{e}_L$ is the $L$-th standard basis vector.

We show that the total variation distance between $\mathcal{Q}(\mathbf{U}^n)$ and $\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})$ is at least a constant. This time, we argue that the former is more likely to output a "small" vector than the latter. Specifically,

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}(\mathbf{U}^n), \mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})) \\
&\geq \mathbb{P}\left[\|\mathcal{Q}(\mathbf{U}^n)\|_\infty \leq \alpha\right] - \mathbb{P}\left[\|\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})\|_\infty \leq \alpha\right] \\
&= \mathbb{P}\left[\|\mathcal{Q}(\mathbf{U}^n) - \mathbb{E}\left[\mathbf{U}\right]\|_\infty \leq \alpha\right] - \mathbb{P}\left[\|\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})\|_\infty \leq \alpha\right] && (\mathbb{E}\left[\mathbf{U}\right] = \vec{0}) \\
&\geq \frac{99}{100} - \mathbb{P}\left[\|\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})\|_\infty \leq \alpha\right] \\
&\geq \frac{99}{100} - \mathbb{P}\left[\|\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha}) - \mathbb{E}\left[\mathbf{D}_{d,L,B,\alpha}\right]\|_\infty > \alpha\right] && (\|\mathbb{E}\left[\mathbf{D}_{d,L,B,\alpha}\right]\|\infty = 2\alpha) \\
&\geq \frac{99}{100} - \frac{1}{100} = \frac{49}{50}
\end{aligned}$$

From Theorem 3.3.11, we conclude that $n = \Omega\left(\sqrt{d}/\alpha\varepsilon\right)$. □

**Theorem A.5.5** (Restatement of Theorem 3.3.18)**.** *If $\mathcal{D}$ is an $(\varepsilon, \delta)$-robustly shuffle private protocol that solves $(d, 1, \alpha)$-sparse mean estimation and $\delta \log d/\delta \ll \alpha^2\varepsilon^2/d$, then its sample complexity is $n = \Omega(\sqrt{d}/\alpha\varepsilon)$.*

*Proof.* As before, let $\mathbf{D}_{d,L,B,\alpha}$ denote a distribution chosen uniformly at random from $\mathcal{D}_{d,1,\alpha}$. Assume $\Pi$ is a shuffle-model protocol that solves $(d, 1, 2\alpha/9)$-sparse mean estimation. We show that $\mathcal{Q}_{\mathcal{P}}$ distinguishes between $\mathbf{U}^{n/3}$ and $\mathbf{D}^{n/3}_{d,L,B,\alpha}$.

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}_{\mathcal{P}}(\mathbf{U}^{n/3}), \mathcal{Q}_{\mathcal{P}}(\mathbf{D}^{n/3}_{d,L,B,\alpha})) \\
&\geq \mathbb{P}\left[\|\mathcal{Q}_{\mathcal{P}}(\mathbf{U}^{n/3})\|_\infty \leq 2\alpha/9\right] - \mathbb{P}\left[\|\mathcal{Q}_{\mathcal{P}}(\mathbf{D}^{n/3}_{d,L,B,\alpha})\|_\infty \leq 2\alpha/9\right] \\
&\geq \mathbb{P}\left[\|\Pi(\mathbf{U}^n)\|_\infty \leq 2\alpha/9\right] - \mathbb{P}\left[\|\Pi(\mathbf{D}^n_{d,L,B,2\alpha/9})\|_\infty \leq 2\alpha/9\right] - \frac{1}{6} && \text{(Lemma 3.2.2)} \\
&\geq \frac{49}{50} - \frac{1}{6} = \frac{61}{75}
\end{aligned}$$

The third inequality comes from repeating the analysis in the proof of Theorem 3.3.17. As before, we invoke Theorem 3.3.11 to conclude that $n = \Omega(\sqrt{d}/\alpha\varepsilon)$. The claimed theorem follows from rescaling $\alpha$ and $d$. □

### A.5.3 Parity Release

**Theorem A.5.6** (Restatement of Theorem 3.3.20)**.** *If $\mathcal{Q}$ is an $(\varepsilon, \delta)$-internally private algorithm that releases width-$k$ parities with error $\alpha$ and $\delta \log \binom{d}{\leq k}/\delta \ll \alpha^2\varepsilon^2/\binom{d}{\leq k}$, then its sample complexity is $n = \Omega(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon)$.*

*Proof.* Analogous to the previous proofs, let $\mathbf{D}_{d,L,B,\alpha}$ denote a distribution chosen uniformly at random from the family $\mathcal{D}_{d,k,\alpha}$. We show that the total variation distance between $\mathcal{Q}(\mathbf{U}^n)$ and $\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})$ is at least a constant. This time, we argue that the former is more likely to output a function bounded by $\alpha$ than the

latter. Specifically,

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}(\mathbf{U}^n), \mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})) \\
&\geq \mathop{\mathbb{P}}_{F\sim\mathcal{Q}(\mathbf{U}^n)}[\forall \ell \subseteq [d], |\ell| \leq k \,|F(\ell)| \leq \alpha] - \mathop{\mathbb{P}}_{F\sim\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})}[\forall \ell \subseteq [d], |\ell| \leq k \,|F(\ell)| \leq \alpha] \\
&\geq \frac{99}{100} - \mathop{\mathbb{P}}_{F\sim\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})}[\forall \ell \subseteq [d], |\ell| \leq k \,|F(\ell)| \leq \alpha] \qquad \text{(A.5)} \\
&\geq \frac{99}{100} - \mathop{\mathbb{P}}_{F\sim\mathcal{Q}(\mathbf{D}^n_{d,L,B,\alpha})}[\forall \ell \subseteq [d], |\ell| \leq k \,|F(\ell) - 2\alpha| > \alpha] \\
&\geq \frac{99}{100} - \frac{1}{100} \qquad \text{(A.6)} \\
&= \frac{49}{50}
\end{aligned}$$

Inequality (A.5) follows from the fact that $\forall \ell, b \mathop{\mathbb{E}}_{x\sim\mathbf{U}}\left[\prod_{j\in\ell} x_j\right] = 0$ and the correctness of $M$. Meanwhile (A.6) follows from the fact that $\forall \ell, b \mathop{\mathbb{E}}_{x\sim\mathbf{D}_{d,\ell,b,\alpha}}\left[\prod_{j\in\ell} x_j\right] = 2\alpha b$ and the correctness of $M$. From Theorem 3.3.11, we conclude the claimed lower bound on $n$. □

**Theorem A.5.7** (Restatement of Theorem 3.3.21)**.** *If $\mathcal{D}$ is an $(\varepsilon, \delta)$-robustly shuffle private protocol that releases width-$k$ parities with error $\alpha$ and $\delta \log\binom{d}{\leq k}/\delta \ll \alpha^2\varepsilon^2/\binom{d}{\leq k}$, then its sample complexity is $n = \Omega(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon)$.*

*Proof.* Again, let $\mathcal{Q}_{\mathcal{P}}$ denote the $(\varepsilon, \delta)$-internally private algorithm given by Lemma 3.2.2. We show that $\mathcal{Q}_{\mathcal{P}}$ distinguishes between $\mathbf{U}^{n/3}$ and $\mathbf{D}^{n/3}_{d,L,B,\alpha}$.

$$\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathcal{Q}_{\mathcal{P}}(\mathbf{U}^{n/3}), \mathcal{Q}_{\mathcal{P}}(\mathbf{D}^{n/3}_{d,L,B,\alpha})) \\
&\geq \mathop{\mathbb{P}}_{F\sim\mathcal{Q}_{\mathcal{P}}(\mathbf{U}^{n/3})}[\forall \ell \subseteq [d], |\ell| \leq k \,|F(\ell)| \leq 2\alpha/9] - \mathop{\mathbb{P}}_{F\sim\mathcal{Q}_{\mathcal{P}}(\mathbf{D}^{n/3}_{d,L,B,\alpha})}[\forall \ell \subseteq [d], |\ell| \leq k \,|F(\ell)| \leq 2\alpha/9] \\
&\geq \mathop{\mathbb{P}}_{F\sim\Pi(\mathbf{U}^n)}[\forall \ell \subseteq [d], |\ell| \leq k \,|F(\ell)| \leq 2\alpha/9] - \mathop{\mathbb{P}}_{F\sim\Pi(\mathbf{D}^n_{d,L,B,2\alpha/9})}[\forall \ell \subseteq [d], |\ell| \leq k \,|F(\ell)| \leq 2\alpha/9] - \frac{1}{6} \\
&\geq \frac{49}{50} - \frac{1}{6} = \frac{61}{75}
\end{aligned}$$

The third inequality comes from repeating the analysis in the proof of Theorem 3.3.20. As before, we invoke Theorem 3.3.11 to conclude the claimed lower bound on $n$. □