Differential Privacy in the Shuffle Model: A Survey of Separations

Albert Cheu

August 18, 2020

Abstract

Classical work in differential privacy operates in extremes of trust assumptions: either all users give their data to a single party or they have no trust in any party. The shuffle model posits an intermediary level of trust in hopes of gaining an intermediary level of accuracy. This survey gives an overview of results in the shuffle model which validate that trade-off.

1 Introduction

Many differentially private algorithms operate in the *central model*, also known as the trusted curator model. Here, a single analyzer has data on all users in question and the objective is to compute in a manner that is distributionally insensitive to any one user's datum. But the fact that all users give their data to one party means that there is a single point of failure: if the analyzer is corrupt or otherwise breached, the privacy of all users will be violated.

There are a number of ways to model weaker trust in the analyzer, chief among them being the *local model*. Here, the dataset is a distributed object where each user holds a single row. To preserve their own privacy, each user executes a randomizing function on their data point and submits the resulting outputs to the analyzer. Due to the severely weakened signal, there are a number of lower bounds on the error of locally private protocols that strongly separate the local model from the central model [11, 14, 10, 1, 29]. Additionally, locally private protocols are particularly susceptible to *manipulation attacks*: by sending carefully distributed messages, malicious users can skew tests and estimates of distributions beyond simply changing the input of the protocol [15].

These negative results motivate the exploration of alternative models. One tactic is to apply secure multi-party computation (MPC) to simulate arbitrary central model algorithms (see e.g. Dwork et al. [17]). However, current methods of simulating generic circuits impose large costs in terms of computation and communication and are difficult to scale and maintain. Another tactic is to identify a common operation in central model algorithms and then implement it with a cryptographic primitive, possibly with MPC.

Because statistical computations tend to be commutative (e.g. mean), it is without loss of accuracy to apply a uniformly random permutation. On the other hand, if the values being permuted are sent by users, privacy can only be enhanced because an adversary cannot identify the source of any value. To be more specific, users in the *shuffle model* send randomized messages to a *shuffler* which uniformly permutes the messages before sending to the analyzer. It is this shuffled collection of messages that needs to satisfy differential privacy: altering one user's data point must not greatly change the distribution of the shuffled messages.

To have differential privacy, a user needs only to trust that the shuffler acts as intended and sufficiently many of their peers follow the protocol. A more detailed description of a shuffler and its implementation details appears in the work of Bittau et al. [12].

The goal of this survey is to summarize the surge of work in the shuffle model, placing emphasis on *separation results*. To this end, the survey is structured as follows. We first establish the requisite privacy and model definitions. Next we contrast local model lower bounds with shuffle model upper bounds: there are four known problems for which additive error and sample complexity are much lower in the shuffle model. We close the main body with techniques to show that the shuffle model (under natural constraints) is weaker than the central model. In Appendix A, we present results by Balle et al. [7, 8] and Ghazi et al. [25, 24, 23] concerning *communication-efficient shuffle protocols*. These inspire confidence in the practicality of the model.

2 Preliminaries

We will use the notation $[k] = \{1, 2, \ldots, k\}$, $\mathbb{N} = \{1, 2, \ldots\}$. A dataset $\vec{x} \in \mathcal{X}^n$ is an ordered tuple of n rows where each row is drawn from a data universe $\mathcal{X}$ and corresponds to the data of one 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 $\vec{x} \sim \vec{x}'$.

Definition 1 (Differential Privacy [18])**.** An algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{Z}$ satisfies (ε, δ)*-differential privacy* if, for every pair of neighboring datasets $\vec{x}$ and $\vec{x'}$ and every subset $T \subset \mathcal{Z}$,

$$\mathbb{P}\left[\mathcal{M}(\vec{x}) \in T\right] \leq e^{\varepsilon} \cdot \mathbb{P}\left[\mathcal{M}(\vec{x}') \in T\right] + \delta.$$

When $\delta > 0$, we say $\mathcal{M}$ satisfies *approximate* differential privacy. When $\delta = 0$, $\mathcal{M}$ satisfies *pure* differential privacy and we omit the δ parameter.

Because this definition assumes that the algorithm $\mathcal{M}$ has "central" access to compute on the entire raw dataset, we sometimes call this *central* differential privacy. Two common facts about differentially privacy will be useful; proofs appear in Chapter 2 of the survey of Dwork and Roth [20]. First, privacy is preserved under post-processing.

Fact 2. *For (ε, δ)-differentially private algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{Z}$ and arbitrary random function $f : \mathcal{Z} \to \mathcal{Z}'$, $f \circ \mathcal{M}$ is (ε, δ)-differentially private.*

This means that any computation based solely on the output of a differentially private function does not affect the privacy guarantee. Second, privacy composes neatly.

Fact 3. *For $(\varepsilon_1, \delta_1)$-differentially private $\mathcal{M}_1$ and $(\varepsilon_2, \delta_2)$-differentially private $\mathcal{M}_2$, $\mathcal{M}_3$ defined by $\mathcal{M}_3(\vec{x}) = (\mathcal{M}_1(\vec{x}), \mathcal{M}_2(\vec{x}))$ is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differentially private.*

One useful centrally private algorithm is the *binomial mechanism*.

Lemma 4 (Binomial Mechanism [17, 24])**.** *Let $f : \mathcal{X}^n \to \mathbb{Z}$ be a 1-sensitive function, i.e. $|f(\vec{x}) - f(\vec{x}')| \leq 1$ for all neighboring datasets $\vec{x}, \vec{x}' \in \mathcal{X}^n$. Fix any $\ell \in \mathbb{N}$, $p \in (0, 1)$, $\varepsilon > 0$, and $\delta \in (0, 1)$ such that*

$$\ell \cdot \min(p, 1-p) \geq 10 \cdot \left(\tfrac{e^{\varepsilon}+1}{e^{\varepsilon}-1}\right)^2 \cdot \ln\left(\tfrac{2}{\delta}\right).$$

The algorithm that samples $\eta \sim \mathbf{Bin}(\ell, p)$ and outputs $f(\vec{x}) + \eta$ is (ε, δ)-differentially private. The error is $O(\frac{1}{\varepsilon}\sqrt{\log \frac{1}{\delta}})$.

We will also use the *geometric mechanism*. Let $\mathbf{SG}(\varepsilon)$ denote the symmetric geometric distribution with parameter ε. For every $v \in \mathbb{Z}$, it has mass $e^{-\varepsilon|v|} \cdot \frac{e^{\varepsilon}-1}{e^{\varepsilon}+1}$. It is a discrete analogue of the Laplace distribution with mean 0.

Lemma 5 (Geometric Mechanism [27])**.** *Let $f : \mathcal{X}^n \to \mathbb{Z}$ be a 1-sensitive function, i.e. $|f(\vec{x}) - f(\vec{x}')| \leq 1$ for all neighboring datasets $\vec{x}, \vec{x}' \in \mathcal{X}^n$. Then for $\varepsilon > 0$, the algorithm that samples $\eta \sim \mathbf{SG}(\varepsilon)$ then outputs $f(\vec{x}) + \eta$ is ε-differentially private. The error is $O(\frac{1}{\varepsilon})$.*

2.1 The Local Model

We first establish the local model. Here, the dataset is 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 6 (Local Model [32, 22])**.** 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 n and an arbitrary amount of public randomness.

Suppose an adversary wishes to recover some information about user i. In this model, the adversary's view is limited to the output of $\mathcal{R}(x_i)$ so it is natural to impose the privacy constraint on $\mathcal{R}$.

Definition 7 (Local Differential Privacy [18, 30])**.** A protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is (ε, δ)-*local differentially private* if $\mathcal{R}$ is (ε, δ)-differentially private. The privacy guarantee is over the internal randomness of the users' randomizers and not the public randomness of the protocol.

2.2 The Shuffle Model

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: users execute $\mathcal{R}$ on their data to produce messages as before, but now users trust a party to perform a secure shuffle on the messages. That is, an adversary's view is limited to a uniformly random permutation of the messages, so no message can be linked back to its sender. Intuitively, whatever privacy guarantee is granted by $\mathcal{R}$ is *amplified* by this anonymity: to learn about x_i, an adversary has to not only recover information from one noisy message y_i but somehow identify the target message inside a vector $\vec{y}$ of n messages. The "amplification-by-shuffling" lemmas due to Erlingsson et al. and Balle et al. quantify how well the privacy parameters are improved [21, 9].

We will abstract away how a shuffler is implemented. Aside from simplifying the analysis, the elision is natural because there are a number of existing shuffling methods with minimum trust assumptions. For example, a mixnet is a sequence of computational nodes that each apply a uniform permutation and pass on to the next node; we only need one honest node to have the desired output. Another, more detailed description of a shuffler appears in the work of Bittau et al. [12].

There are two generalizations of the preliminary model. In one, we *shuffle the dataset itself* prior to generating messages instead of shuffling the messages. In addition to single-message protocols, this generalization also encompasses *sequentially interactive* protocols, where the randomizer of user i adapts to the messages of users $1, \ldots, i-1$. The amplification-by-shuffling lemma in [21] holds in this version of the shuffle model.

We focus on the generalization where *each user can send any number of (atomic) messages at once* to the shuffler. The shuffling prevents messages from the same sender from being linked with one another. Although interactivity is absent, we are still able to find strong separations with the local model.

Definition 8 (Shuffle Model [12, 16])**.** A protocol $\mathcal{P}$ in the *shuffle model* consists of three randomized algorithms:

- A *randomizer* $\mathcal{R} : \mathcal{X} \to \mathcal{Y}^*$ mapping data to (possibly variable-length) vectors. The length of the vector is the number of messages sent. If, on all inputs, the probability of sending m messages is 1, then we have an *m-message protocol*.

- A *shuffler* $\mathcal{S} : \mathcal{Y}^* \to \mathcal{Y}^*$ that concatenates message vectors and then applies a uniformly random permutation to the messages.
- 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 on input $\vec{x} \in \mathcal{X}^n$ as

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

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

With this setup, we use the following definition of shuffle differential privacy.

Definition 9 (Shuffle Differential Privacy [16])**.** A protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is (ε, δ)-*shuffle differentially private* if, for all $n \in \mathbb{N}$, the algorithm $(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}) := \mathcal{S}(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$ is (ε, δ)-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 9 assumes all users follow the protocol. Ideally, the definition should account for malicious users that aim to make the protocol less private: the parameters should degrade smoothly with the number of malicious users. A simple attack is for such users to drop out: for $\gamma \leq 1$, let $\mathcal{S} \circ \mathcal{R}^{\gamma n}$ denote the case where only γn out of n users execute $\mathcal{R}$. Because the behavior of the randomizer may depend on n, $\mathcal{S} \circ \mathcal{R}^n$ may satisfy a particular level of differential privacy but $\mathcal{S} \circ \mathcal{R}^{\gamma n}$ may not.[1] This motivates a *robust* variant of shuffle privacy.

Definition 10 (Robust Shuffle Differential Privacy [6])**.** Fix any $\gamma \in (0, 1]$. A protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is $(\varepsilon, \delta, \gamma)$-*robustly shuffle differentially private* if, for all $n \in \mathbb{N}$ and $\gamma' \in \{1/n, 2/n, \ldots, 1\} \cap [\gamma, 1]$, the algorithm $\mathcal{S} \circ \mathcal{R}^{\gamma' n}$ is (ε, δ)-differentially private. In other words, $\mathcal{P}$ guarantees (ε, δ)-shuffle privacy whenever at least a γ fraction of users follow the protocol.

As with generic shuffle differential privacy, we often shorthand this as "robust shuffle privacy." Note that we define robustness with regard to privacy rather than accuracy. A robustly shuffle private 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; we will state our accuracy guarantees under the assumption that all users follow the protocol.

We emphasize that robustness is not immediately implied by the basic form of shuffle privacy in Definition 9. Indeed, Appendix B details protocols that satisfy shuffle privacy but are not robust to drop-outs.

3 Separations between Local & Shuffle Privacy

In this section, we will introduce four problems. For each problem, we will state a lower bound in the local model and then describe a protocol in the shuffle model that breaks through that bound. To simplify the presentation, we will assume $\varepsilon < 1$ and $\delta = O(1/\text{poly}(n))$.

[1] Note that, with respect to differential privacy, dropping out is "the worst" malicious users can do. This is because adding messages from malicious users to those from honest users is a post-processing of $\mathcal{S} \circ \mathcal{R}^{\gamma n}$. If $\mathcal{S} \circ \mathcal{R}^{\gamma n}$ is already differentially private for the outputs of the γn users alone, then differential privacy's resilience to post-processing (Fact 2) ensures that adding other messages does not affect this guarantee. Hence, it is without loss of generality to focus on drop-out attacks.

3.1 Binary Sums

In this setting, each user i has a bit $x_i \in \{0,1\}$ and the objective is to compute the sum. Dating back to Warner [32], *randomized response* is the canonical local protocol for this problem. The randomizer is below:

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

Let y_i be the message sent by user i. Due to subsampling and noise addition, the expected value of $\sum y_i$ is $(1-p)\cdot\sum x_i + np/2$. The analyzer will re-center and re-scale to obtain an unbiased estimator:

$$\begin{aligned} \mathcal{A}_{\mathsf{RR}}(\vec{y}) &:= \frac{1}{1-p}\left(\sum y_i - np/2\right) \\ \mathbb{E}\left[\mathcal{A}_{\mathsf{RR}}(\vec{y})\right] &= \frac{1}{1-p}\cdot\left(\mathbb{E}\left[\sum y_i\right] - np/2\right) \\ &= \sum x_i \end{aligned}$$

Setting $p \leftarrow 2/(e^\varepsilon+1)$ suffices for ε-local privacy but incurs an additive error of $O(\frac{1}{\varepsilon}\sqrt{n})$. This is optimal.

Theorem 11 (Beimel et al. [11] & Chan et al. [14])**.** *Let $\mathcal{P}$ be an (ε,δ)-locally private protocol. If $\mathcal{P}$ computes binary sums up to additive error α with constant probability, then $\alpha = \Omega(\frac{1}{\varepsilon}\sqrt{n})$.*

Note that $\mathcal{P}_{\mathsf{RR}} := (\mathcal{R}_{\mathsf{RR}}, \mathcal{A}_{\mathsf{RR}})$ can also be interpreted as a single-message shuffle protocol. Cheu et al. [16] show that the parameter p can be chosen such that RR achieves robust shuffle privacy while also avoiding error that scales polynomially with n.

Theorem 12 (Cheu et al. [16])**.** *Randomized response $\mathcal{P}_{\mathsf{RR}} = (\mathcal{R}_{\mathsf{RR}}, \mathcal{A}_{\mathsf{RR}})$ is a single-message protocol that is $(\varepsilon, 3\delta^\gamma, \gamma)$-robustly shuffle private for any $\gamma \in (0,1]$. It computes binary sums up to additive error $O(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}})$ with constant probability.*

Proof. If $n > 80\cdot(\frac{e^\varepsilon+1}{e^\varepsilon-1})^2\ln\frac{1}{\delta}$, we set p to $\frac{40}{n}\cdot(\frac{e^\varepsilon+1}{e^\varepsilon-1})^2\ln\frac{1}{\delta}$. If n is smaller, p must take a different form and the analysis will naturally change; we omit this technicality for neatness (refer to [16] for more details).

Robust privacy: The shuffler's output is $\vec{y} \leftarrow (\mathcal{S}\circ\mathcal{R}_{\mathsf{RR}}^{\gamma n})(\vec{x})$, a vector of bits whose order is chosen uniformly at random. Observe that the permutation does not contain information about the data: given oracle access to the sum $\sum y_i$, a privacy adversary can exactly simulate the distribution of $\vec{y}$. Hence it suffices to ensure that $\sum y_i$ is differentially private.

Let H denote the set of users who report messages y_i from $\mathbf{Ber}(1/2)$. Notice that $\sum y_i = \sum_{i\notin H} x_i + \sum_{i\in H} y_i$ so that $\sum y_i \sim \sum_{i\notin H} x_i + \mathbf{Bin}(|H|, 1/2)$. If $|H| \geq 20\cdot(\frac{e^\varepsilon+1}{e^\varepsilon-1})^2\ln\frac{1}{\delta^\gamma}$, then we have $(\varepsilon, 2\delta^\gamma)$ privacy by the binomial mechanism (Lemma 4).

We will argue that this bound on $|H|$ holds with probability $1-\delta^\gamma$ for our choice of p, so we have $(\varepsilon, 3\delta^\gamma)$ privacy overall. The expected value of $|H|$ is $\mathbb{E}[|H|] = \gamma n\cdot p = 40\cdot(\frac{e^\varepsilon+1}{e^\varepsilon-1})^2\ln\frac{1}{\delta^\gamma}$ which is sufficiently large to invoke a Chernoff bound: $|H| \geq \frac{1}{2}\cdot\mathbb{E}[|H|] = 20\cdot(\frac{e^\varepsilon+1}{e^\varepsilon-1})^2\ln\frac{1}{\delta^\gamma}$ with probability $\geq 1-\delta^\gamma$.

Accuracy: We bound the protocol's error under the assumption that all users are honest ($\gamma = 1$). Recall that the output of the protocol is $\frac{1}{1-p}(\sum y_i - np/2)$. By a Chernoff bound, we have that $\sum y_i - np/2$ is within $O(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}})$ of its expectation. And because $\frac{1}{1-p} < 2$, the error of the unbiased estimator is $O(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}})$. □

We remark that there are other shuffle protocols for binary sums with low error; Table 1 presents their most salient features.

Table 1: Shuffle protocols for binary sums. Each message is one bit. "w.c.p." denotes a bound that holds with constant probability over the randomness of all users.

Given name	Error w.c.p.	No. Messages per User	Advantage over RR	Source
RR	$O(\frac{1}{\varepsilon} \cdot \sqrt{\log \frac{1}{\delta}})$	1	—	[16]
ZSUM	$O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$	2	If sum is 0, estimate is 0	[5]
—	$O(\frac{1}{\varepsilon^{3/2}} \cdot \sqrt{\log \frac{1}{\varepsilon}})$	$O(\frac{1}{\varepsilon} \log n)$	$\delta = 0$	[23]
—	$O(\frac{1}{\varepsilon} \cdot \sqrt{\log \frac{1}{\delta}})$	$O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$ w.c.p.	noise symmetry	[6]

3.2 Histograms

In this setting, each user has one value in the set $[k]$. The objective is to privately estimate the count of each element j. To be precise, the output must be a vector $(\tilde{c}_1, \ldots, \tilde{c}_k)$ minimizing ℓ_∞ error with $(c_1, \ldots, c_k)$ where c_j is the count of j in the input dataset. This must grow with k under local privacy:

Theorem 13 (Bassily & Smith [10])**.** *Let $\mathcal{P}$ be an (ε, δ)-locally private protocol. If $\mathcal{P}$ reports a histogram that has ℓ_∞ error α with constant probability, then $\alpha = \Omega(\frac{1}{\varepsilon}\sqrt{n \log k})$*

In contrast, it is possible to have error independent of k under robust shuffle privacy:

Theorem 14 (Balcer et al. [5, 6])**.** *There is a $2k$-message histogram protocol that satisfies $(2\varepsilon, 4\delta^\gamma, \gamma)$-robust shuffle privacy for any $\gamma \in (0, 1]$. Its estimate has ℓ_∞ error $O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$ with constant probability.*

Proof. A simple way to obtain a private histogram is to count each of the k elements with a binary sum protocol, then perform a union bound. The k protocol executions can be done in parallel by having each user label their messages. To be precise, let $\mathcal{R}_j$ be a binary sum randomizer that counts the occurrences of j. If $\mathcal{R}_j(x_i)$ outputs messages a and b, user i reports the labeled messages $\langle j, a\rangle$ and $\langle j, b\rangle$. The analyzer can feed the messages for counting j into the corresponding function $\mathcal{A}_j$.

A side-effect of this reduction approach is that the union bound may create a dependence on k. For example, if we use RR for frequency estimation, the ℓ_∞ error after the union bound has a $\sqrt{\log k}$ term. But we can avoid this dependence by using ZSUM, a binary sum protocol which guarantees noiseless estimation when the input is $(0, \ldots, 0)$. As such, the elements with nonzero frequency will be the only ones with noisy estimates, so the union bound is over $\leq n$ executions instead of k.

We will sketch the protocol ZSUM for target privacy parameters $(\varepsilon, 2\delta^\gamma)$ so that simple composition will give our overall privacy guarantee. We begin with the two-message randomizer, which takes parameter $r \in (0, 1)$.

$$\mathcal{R}_{\mathsf{ZSUM}}(x_i) := (x_i, \mathbf{Ber}(r))$$

Robust Privacy: As with RR, the number and order of messages produced by $(\mathcal{S} \circ \mathcal{R}^n_{\mathsf{ZSUM}})$ is data independent, so it suffices to prove privacy of the message sum $\sum y_{i,1} + y_{i,2}$. But this quantity is exactly $\sum x_i + \eta$, where η is drawn from the distribution $\mathbf{Bin}(n, r)$. By Lemma 4, it suffices to choose $r = 1 - \frac{10}{n} \cdot \left(\frac{e^\varepsilon + 1}{e^\varepsilon - 1}\right)^2 \cdot \ln \frac{1}{\delta}$ for $(\varepsilon, 2\delta^\gamma)$ privacy.

Accuracy: Now we define the analyzer $\mathcal{A}_{\mathsf{ZSUM}}$.

$$\mathcal{A}_{\mathsf{ZSUM}}(\vec{y}) := \begin{cases} 0 & \text{if } \sum y_{i,1} + y_{i,2} \leq n \\ \sum y_{i,1} + y_{i,2} - nr & \text{otherwise} \end{cases}$$

First consider the case where $\sum x_i = 0$. Because $\eta \sim \mathbf{Bin}(n, r)$ has maximum value n, $\mathbb{P}\left[\sum y_{i,1} + y_{i,2} \le n\right] = 1$ so there is zero error.

Now consider the case where $\sum x_i = 0$. We can use a Chernoff bound to argue that $|\eta - nr| = O(\sqrt{n(1-r)\log n})$ with probability $1/10n$. If we do not truncate, subtracting nr removes bias so that error has magnitude $O(\sqrt{n(1-r)\log n}) = O(\frac{1}{\varepsilon}\log\frac{1}{\delta})$. Otherwise, error is exactly $\sum x_i$. But truncation will not occur when $\sum x_i = \Omega(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$: in this case, $\sum x_i + \eta > \sum x_i + n - O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ so that $\sum y_{i,1} + y_{i,2} > n$.

When $n < 10 \cdot \left(\frac{e^\varepsilon+1}{e^\varepsilon-1}\right)^2 \cdot \ln\frac{1}{\delta}$, our choice of r is not a valid probability. In this case, each user can deterministically send the bits $0, 0$ to ensure $(0,0)$-privacy. $\mathcal{A}_{\mathsf{ZSUM}}$ will report 0 with probability 1, which has error $\sum x_i \le n = O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$. □

We remark that the number of messages per user can be changed to $O(n \cdot k^{1/T})$ for an arbitrary integer constant $T > 1$ at the price of inflating the error by a factor of $\approx T^2$. We make this argument in the Appendix.

3.3 Uniformity testing

The results so far concern protocols that make no assumption about where user data comes from. We now transition to problems where each user has one i.i.d. sample from some probability distribution $\mathbf{D}$ over $[k]$. In the case of *α-uniformity testing*, the objective is to report "uniform" with probability $2/3$ when $\mathbf{D} = \mathbf{U}$ and "not uniform" with probability $2/3$ when $\|\mathbf{D} - \mathbf{U}\|_{\mathrm{TV}} > \alpha$. The minimum number of users needed to ensure those two conditions hold is the *sample complexity* of the protocol. Under local privacy, this must scale at least linearly with k.

Theorem 15 (Acharya et al. [1])**.** *If an ε-locally private protocol performs α-uniformity testing, then its sample complexity is $\Omega(k/\alpha^2\varepsilon^2)$.*

But under robust shuffle privacy, the sample complexity is smaller by at least a polynomial factor.

Theorem 16 (Balcer et al. [6])**.** *There is a multi-message protocol that satisfies $(\varepsilon, 4\delta^\gamma, \gamma)$-robust shuffle privacy for any $\gamma \in (0,1]$ and solves α-uniformity testing with sample complexity*

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

Proof. We note that we will take $n \sim \mathbf{Pois}(m)$ and upper bound m. This "Poissonization" has the effect of making the random variables $c_1, \ldots, c_k$ mutually independent, which simplifies the analysis.

Cai et al. [13] give a recipe for private uniformity testing under Poissonization. First, compute a private histogram $(\tilde{c}_1, \ldots, \tilde{c}_k)$. Then, compute the test statistic

$$Z'(\tilde{c}_1, \ldots, \tilde{c}_k) := \frac{k}{m}\sum_{j=1}^{k}(\tilde{c}_j - m/k)^2 - \tilde{c}_j \tag{1}$$

The final step is to prove that this statistic is small when the data distribution is uniform but large when it is α-far from uniform, which means we can distinguish the two cases with a threshold test.

Amin et al. [4] give the following procedure to analyze Z'. If we let η_j be the noise in $\tilde{c}_j$ introduced by

privacy, then we rewrite Z' as

$$\begin{aligned}(1) &= \frac{k}{m}\sum_{j=1}^{k}(c_j+\eta_j-m/k)^2-c_j-\eta_j \\ &= \underbrace{\frac{k}{m}\sum_{j=1}^{k}(c_j-m/k)^2-c_j}_{Z}+\underbrace{\frac{k}{m}\sum_{j=1}^{k}\eta_j^2}_{A}+\underbrace{\frac{2k}{m}\sum_{j=1}^{k}\eta_j\cdot(c_j-m/k)}_{B}-\underbrace{\frac{k}{m}\sum_{j=1}^{k}\eta_j}_{C}\end{aligned}$$

Analysis in Acharya et al. [2] imply bounds on term Z in the two relevant cases: there is a constant t and a function $f(\alpha, m)$ such that

- when $\|\mathbf{D}-\mathbf{U}\|_{\mathrm{TV}} > \alpha$, $Z > t \cdot f(\alpha, m)$ with constant probability
- when $\mathbf{D} = \mathbf{U}$, $Z \le f(\alpha, m)$ with constant probability

It will suffice to prove the two statements below:

(i) when $\|\mathbf{D}-\mathbf{U}\|_{\mathrm{TV}} > \alpha$, $A+B+C > 0$ with constant probability

(ii) when $\mathbf{D} = \mathbf{U}$, $A+B+C < (t-1)\cdot f(\alpha, m)$ with constant probability

Balcer et al. [6] describe a binary sum protocol which produces estimates with zero-mean symmetric noise. A corollary is that there is a private histogram protocol where each η_j is an independent sample from zero-mean symmetric noise. (i) immediately follows. (ii) follows from Chebyshev's inequality and the moments of η_j. □

3.4 Pointer-Chasing

The *pointer chasing problem* is denoted $\mathrm{PC}(d, \ell)$ where $d, \ell \in \mathbb{N}$. A problem instance is a set $\{(1, \vec{a}), (2, \vec{b})\}$, where $\vec{a}$ and $\vec{b}$ are permutations of $[\ell]$. A protocol *solves* $\mathrm{PC}(d, \ell)$ *with sample complexity* n if, given n independent samples drawn uniformly with replacement from any problem instance $\{(1, \vec{a}), (2, \vec{b})\}$, it outputs the d-th integer in the sequence $a_1, b_{a_1}, a_{b_{a_1}} \ldots$ with constant probability.

The sample complexity of $\mathrm{PC}(d, \ell)$ under local privacy must scale at least linearly with ℓ.

Theorem 17 (Joseph et al. [29])**.** *If an* (ε, δ)*-locally private protocol solves* $\mathrm{PC}(2, \ell)$ *with sample complexity* n *then* $n = \Omega(\ell)$*.*

In stark contrast, the sample complexity under shuffle privacy is *independent of* ℓ:

Theorem 18 (Balcer & Cheu [5])**.** *There is a* $8 \cdot (\ell!)^2$*-message protocol that satisfies* $(2\varepsilon, 4\delta^\gamma, \gamma)$*-robust shuffle privacy for any* $\gamma \in (0, 1]$ *and solves* $\mathrm{PC}(2, \ell)$ *with sample complexity* $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$*.*

Proof. Let $\pi(\ell)$ denote all permutations of $[\ell]$. Observe that the tuples $(1, \vec{a}), (2, \vec{b})$ are elements of the universe $\{1, 2\} \times \pi(\ell)$ which has size $2 \cdot \ell!$. We can solve the problem once we have a protocol that singles out $(1, \vec{a})$ and $(2, \vec{b})$ from the universe with constant probability.

Balcer & Cheu argue that the task of privately identifying $(1, \vec{a})$ and $(2, \vec{b})$ with constant probability is $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$. By a straightforward concentration argument, it suffices to have $O(t)$ samples to ensure $(1, \vec{a})$ and $(2, \vec{b})$ each appear $\ge t+1$ times with constant probability. Taking universe size $k = 4 \cdot (\ell!)^2$, we then use the histogram protocol built atop ZSUM (Theorem 14). When $t = \Omega(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$, it will report nonzero frequencies for $(1, \vec{a})$ and $(2, \vec{b})$ but zero for every other element in the universe. □

4 Separations between Central & Shuffle Privacy

There are results that separate the shuffle model from the central model, but they do not hold for general protocols. Instead, they leverage some structural constraint.

4.1 Single-message Protocols

The first class of lower bounds hold for protocols wherein each user sends exactly one message with probability 1. We begin with a negative result for *bounded-value sums*. Here, a protocol must privately estimate the sum of values in the interval $[0, 1]$.

Theorem 19 (Balle et al. [9]). *If a single-message shuffle protocol satisfies (ε, δ) differential privacy for n users and computes bounded-value sums, then the mean-squared error must be $\Omega(n^{1/3})$.*

In the desirable case of unbiased estimators, this implies an additive error of $\Omega(n^{1/6})$. But in the central model, we can simply use the Laplace mechanism to achieve an error of $O(1/\varepsilon)$.

The techniques used to prove the above are specific to bounded-value sums. A more general technique is to consider what shuffle privacy entails for an arbitrary single-message randomizer.

Lemma 20 (Balcer & Cheu [5]). *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 ε-differential privacy on its own whenever $\mathcal{P}$ is ε-shuffle private.*

This means that under pure differential privacy, the single-message shuffle model is *exactly equivalent* to the local model. So all separations between the central and local models hold here as well.

But it is clear from RR that this exact equivalence does not hold for approximate shuffle privacy. The following lemma accommodates the relaxation.

Lemma 21 (Cheu et al. [16]). *If a single-message protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ satisfies (ε, δ)-shuffle privacy for n users, then $\mathcal{R}$ must satisfy $(\varepsilon + \ln n, \delta)$-differential privacy on its own.*

Thus, we can invoke any local model lower bound that holds for $(\varepsilon + \ln n, \delta)$ privacy. This recipe gives the following lower bound.

Theorem 22 (Ghazi et al. [24]). *Any single-message protocol that satisfies $(1, o(1/n))$-shuffle privacy and outputs histograms with ℓ_∞ 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.

4.2 Robustly Shuffle Private Protocols

The second class of lower bound applies to robustly shuffle private protocols. These lower bounds exploit reductions to *pan-privacy*. First presented by Dwork et al. [19], a pan-private algorithm is an online algorithm such that, for any time t, the joint distribution of the internal state at time t and the final output is differentially private.

We first consider the *α-distinct elements* problem. The objective is to estimate—up to additive error α—the number of elements in $[k]$ that appear in $\vec{x}$ at least once. Balcer et al. [6] show that a robustly shuffle private protocol implies a pan-private algorithm and then invoke lower bounds from the pan-privacy literature.

Lemma 23 (Balcer et al. [6]). *If an $(\varepsilon, \delta, 1/3)$-robustly shuffle private protocol solves α-distinct elements, then there is an (ε, δ)-pan-private algorithm that solves $\alpha + 1$-distinct elements.*

Theorem 24 (Mir et al. [31] and Balcer et al. [6]). *If an (ε, δ)-pan-private algorithm solves α-distinct elements, then $\alpha = \Omega(\sqrt{k/\varepsilon})$.*

Theorem 25 (Balcer et al. [6]). *If an $(\varepsilon, \delta, 1/3)$-robustly shuffle private protocol solves α-distinct elements, then $\alpha = \Omega(\sqrt{k/\varepsilon})$.*

In contrast, the binomial and geometric mechanisms (Lemmas 4 and 5) imply that the error in the central model is *independent* of k.

Proof Sketch of Lemma 23. The pan-private algorithm will first generate $n/3$ copies of 1 and execute $(\mathcal{S} \circ \mathcal{R}^{n/3})$ on that vector. Then the algorithm will read user datum x_1 from the stream, execute $\mathcal{R}(x_1)$, and add the messages to the set. The updates repeat until data from all $n/3$ users are read. The algorithm concludes by passing in $n/3$ more copies of 1 and then executing $\mathcal{A}$ on all messages.

The number of distinct elements is increased by at most 1, so the error is at most $\alpha + 1$. Pan-privacy is ensured because the internal state at any point is a post-processing of $(\mathcal{S} \circ \mathcal{R}^{n/3})$ which is differentially private by robustness. □

We can obtain a lower bound for uniformity testing by using essentially identical steps. This time, the pan-private lower bound comes from Amin et al. [4].

Theorem 26 (Balcer et al. [6]). *If an $(\varepsilon, 0, 1/3)$-robustly shuffle private protocol solves α-uniformity testing, then its sample complexity is $n = \Omega(\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2} + \frac{1}{\alpha\varepsilon})$.*

In contrast, Acharya et al. [3] show that the sample complexity in the central model scales with $\sqrt{k}$.

5 Open Questions

How robust are single-message shuffle protocols? If all $n-1$ of Alice's peers drop out of a (ε, δ)-shuffle private protocol, then she is left with $(\varepsilon'(n), \delta'(n))$ privacy, where we currently know that $\varepsilon'(n) \leq \varepsilon + \ln n$ and $\delta'(n) \leq \delta$ (Lemma 21). It is conceivable that these bounds can be improved. Additionally, we may tighten the amplification-by-shuffling lemmas in [21, 9]. These lemmas state that there are functions $\text{amp}_1, \text{amp}_2$ such that an (ε, δ)-locally private protocol implies an $(\text{amp}_1(\varepsilon, \delta, n), \text{amp}_2(\varepsilon, \delta, n))$-shuffle private protocol. So when $\gamma n - 1$ of Alice's peers are honest, she has $(\text{amp}_1(\varepsilon'(n), \delta'(n), \gamma n), \text{amp}_2(\varepsilon'(n), \delta'(n), \gamma n))$ privacy.

How robust are multi-message shuffle protocols? The fact that each user is performing the same randomization algorithm implies an equal division of labor. Intuitively, this should mean that drop-outs cause a graceful degradation of privacy. But this is not the case for arbitrary privacy parameters in the multi-message setting: as detailed in Appendix B, there are protocols that satisfy a degree of differential privacy (finite ε and $\delta < 1$) for n users but do not for $n-1$ users. We note that the privacy parameters of these counterexamples are large, so it may be the case that robustness is intrinsic for *small* ε, δ.

Are there other techniques to obtain lower bounds for robust shuffle privacy? The current technique is bottlenecked by pan-privacy. For example, there is no known lower bound for uniformity testing under approximate pan-privacy.

How close are robust shuffle privacy and pan-privacy? Given that pan-private algorithms typically maintain a numerical internal state, it is conceivable that we can simulate an arbitrary pan-private algorithm by a robustly shuffle private protocol. On the other hand, the fact that a pan-private algorithm can adapt to its input may imply some separation between the models.

What is (im)possible in the variant of the shuffle model where data is shuffled instead of messages? The amplification-by-shuffling lemma due to Erlingsson et al [21] is one way to obtain positive results but are there other techniques? What problems cannot be solved with few samples, even with the adaptivity permitted by the model?

References

[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, 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.

[3] 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.

[4] Kareem Amin, Matthew Joseph, and Jieming Mao. Pan-private uniformity testing. *CoRR*, abs/1911.01452, 2019.

[5] Victor Balcer and Albert Cheu. Separating local & shuffled differential privacy via histograms. *CoRR*, abs/1911.06879, 2019.

[6] Victor Balcer, Albert Cheu, Matthew Joseph, and Jieming Mao. Connecting robust shuffle privacy and pan-privacy. *CoRR*, abs/2004.09481, 2020.

[7] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Differentially private summation with multi-message shuffling. *arXiv preprint arXiv:1906.09116*, 2019.

[8] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Improved summation from shuffling. *CoRR*, abs/1909.11225, 2019.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Albert Cheu, Adam D. Smith, and Jonathan Ullman. Manipulation attacks in local differential privacy. *CoRR*, abs/1909.09630, 2019.

[16] 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.

[17] 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.

[18] 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.

[19] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *Innovations in Computer Science (ICS)*, 2010.

[20] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 2014.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Private aggregation from fewer anonymous messages. *CoRR*, abs/1909.11073, 2019.

[26] Badih Ghazi, Rasmus Pagh, and Ameya Velingker. Scalable and differentially private distributed aggregation in the shuffled model. *CoRR*, abs/1906.08320, 2019.

[27] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 2012.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

A Communication-efficient Protocols

A.1 Bounded-value Sums

In this setting, users have values in the interval $[0,1]$ and the objective is to privately compute their sum. When each user is willing to send multiple messages, results from Balle et al. [7, 8] and Ghazi et al. [26, 25] show that it is possible to *simulate* the geometric mechanism in the shuffle model.

Theorem 27. *There is an (ε,δ)-shuffle private protocol for bounded-value sums with error $O(\frac{1}{\varepsilon})$ where each user sends $O(\frac{\log(1/\delta)}{\log(n)})$ messages, each consisting of $O(\log n)$ bits.*

Proof Sketch. The first step is to equate a sample from $\mathbf{SG}(\varepsilon)$ with the sum of n samples from another distribution $\mathbf{D}_\varepsilon$. Then we recall a modular arithmetic protocol $\mathcal{P}_{\mathsf{MOD}}$ by Ishai et al. [28] which has the following property: two datasets with the same sum induce two distributions of $(\mathcal{S}\circ\mathcal{R}^n_{\mathsf{MOD}})$ that are δ-close in statistical distance. Now define $\mathcal{R}$ to be the execution of $\mathcal{R}_{\mathsf{MOD}}$ on $y_i \leftarrow x_i + \eta$ where $\eta \sim \mathbf{D}_\varepsilon$.

If an adversary can only recover $\sum y_i$, then we will have ε-differential privacy. And due to the way we use MOD, the output of the shuffler $(\mathcal{S}\circ\mathcal{R}^n)(x_1,\ldots,x_n)$ is δ-close to $(\mathcal{S}\circ\mathcal{R}^n_{\mathsf{MOD}})(\sum y_i, 0,\ldots,0)$. This is enough to ensure approximate differential privacy. The error bound $O(\frac{1}{\varepsilon})$ follows from the fact that we are simulating the geometric mechanism (Lemma 5), as well as the fact that sums exceed the modulus with very low probability.

Refer to Balle et al. [8] and Ghazi et al. [25] for analyses of the message complexity of MOD. □

A.2 Histograms & Range queries

Unlike Balcer & Cheu [5], Ghazi et al [24] strive to minimize communication complexity of histograms rather than error. This is achieved with the count-min and Hadamard response techniques that found success in the local model. Refer to Table 2 for a summary of their results. [24] also explain how to use their protocols in a black-box way to solve the range-query problem. In this setting, data is drawn from $[k]^d$ and the objective is to estimate the number of points in a given rectangle. Refer to Table 3 a summary of the results.

Table 2: Shuffle protocols for histograms. All take $\delta > 0$. We assume $\delta < 1/\log k$ for results from [24] and $\delta \geq e^{-O(n\varepsilon^2)}$ for the result from [5]. The notation $\tilde{O}(\dots)$ suppresses nested logarithms.

Technique	ℓ_∞ Error w.c.p.	No. Messages per User	Bits per Message	Source
k parallel counts	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$	$O(k)$	$O(\log k)$	[5]
Count-Min	$\tilde{O}\left(\frac{1}{\varepsilon}\cdot\sqrt{\log^3 k\cdot\log\frac{1}{\delta}}\right)$	$\tilde{O}\left(\frac{1}{\varepsilon^2}\log^3 k\cdot\log\frac{1}{\delta}\right)$ w.c.p.	$O(\log n+\log\log k)$	[24]
Hadamard	$O\left(\log k+\frac{1}{\varepsilon}\cdot\sqrt{\log k\cdot\log\frac{1}{\varepsilon\delta}}\right)$	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon\delta}\right)$	$O(\log n\log k)$	

Table 3: Shuffle protocols for range queries. All take $\delta > 0$. $n \leq k^d$ for neatness

Technique	Error	Messages per User	Bits per Message
Count-Min	$O(\frac{1}{\varepsilon}\log^{2d+3/2}(k^d)\log\frac{1}{\delta})$	$O(\frac{1}{\varepsilon^2}\log^{3d+3}(k^d)\log\frac{1}{\delta})$ w.c.p.	$O(\log n+\log(d\log k))$
Hadamard	$O(\frac{1}{\varepsilon}\log^{2d+1/2}(k^d)\log\frac{1}{\varepsilon\delta})$	$O(\frac{1}{\varepsilon^2}\log^{2d}(k^d)\log\frac{1}{\varepsilon\delta})$	$O(\log(n)\cdot d\log k)$

A.3 Improving Balcer & Cheu's histogram protocol

Here, we return to the histogram protocol by [5]. For any integer constant $T > 1$, we will show how to change the number of messages sent by each user from $\Theta(k)$ to $\Theta(n\cdot k^{1/T})$. The argument is due to Kobbi Nissim. Recall that $c_j(\vec{x})$ denotes the count of j in the dataset $\vec{x}$.

Let $\hat{k}$ be an arbitrary integer for now. Public randomness chooses c uniformly random hash functions $\{h^{(t)} : [k] \to [\hat{k}]\}_{\forall t\in[c]}$. Each user i loops through $t \in [T]$: compute the hash $x_i^{(t)} \leftarrow h^{(t)}(x_i)$ and then run the histogram randomizer (for universe $[\hat{k}]$) on $x_i^{(t)}$. The shuffler therefore receives $T\cdot 2\hat{k}$ messages from each user. By basic composition, the privacy parameters degrade by only a factor of T. So we rescale ε and δ by T to find that we have maximum error $\alpha = O(\frac{T^2}{\varepsilon^2}\log\frac{T}{\delta})$ with constant probability.

Now we show how the analyzer recovers an approximate histogram for $\vec{x}$. We first define $\vec{x}^{(t)}$ to be the vector of hashes $x_1^{(t)},\dots,x_1^{(t)}$ produced by running $h^{(t)}$ on the data $x_1,\dots,x_n$. Using the messages received by the shuffler, the analyzer will obtain T approximate histograms $\{(\tilde{c}_1^{(t)},\dots,\tilde{c}_{\hat{k}}^{(t)}\}_{t\in[T]}$, where $\tilde{c}_{\hat{j}}^{(t)}$ is an estimate of $c_{\hat{j}}(\vec{x}^{(t)})$. Then, for each $j \in [k]$, reports the minimum of $\tilde{c}_{h^{(t)}(j)}^{(t)}$.

To bound the error of this estimate of $c_j(\vec{x})$, we define E_j to denote the event where there is some t such that, for every $j' \in \vec{x} - \{j\}$, $h^{(t)}(j) \neq h^{(t)}(j')$. When this event occurs, $c_{h^{(t)}(j)}(\vec{x}^{(t)})$ is exactly $c_j(\vec{x})$. If

there is another t' where this condition does not hold, $c_{h^{(t')}(j)}(\vec{x}^{(t')})$ is an *over*estimate of $c_j(\vec{x})$. Thus, the minimum of $c_{h^{(t)}(j)}(\vec{x}^{(t)})$ is exactly the count of j in $\vec{x}$ when E_j occurs. Given that the analyzer can obtain estimates of these counts with error at most α, the minimum of the estimates can only be wrong by α.

It is straightforward to see that $\mathbb{P}\left[\neg E_j\right] = \mathbb{P}\left[\forall t \in [T]\ \exists j' \in \vec{x}\ h_t(j') = h_t(j)\right] \leq (n/\hat{k})^T$. By a union bound, the probability that *any* E_j fails to occur is $\leq k \cdot (n/\hat{k})^T$. If $\hat{k} = n \cdot (100k)^{1/T}$, this failure probability is at most $1/100$.

A.4 Lower Bounds

We close this appendix with a result by Ghazi et al. [23] which states that every communication-bounded shuffle protocol must imply some local protocol with a nontrivial privacy guarantee:

Lemma 28 (Ghazi et al. [23])**.** *Suppose $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ satisfies $O(1)$-shuffle privacy and each user sends m messages of ℓ bits. Then the local randomizer $(\mathcal{S} \circ \mathcal{R}^1)$ satisfies $(0, 1 - 2^{-O(m^2\ell)})$-differential privacy.*

By way of the local model, this implies a lower bound for binary sums:

Corollary 29 (Ghazi et al. [23])**.** *If an m-message shuffle protocol satisfies $O(1)$-differential privacy and computes binary sums up to error $o(\sqrt{n})$, then $m^2\ell = \Omega(\log n)$.*

B Shuffle Protocols with Brittle Privacy

Here, we describe two protocols which satisfy non-trivial shuffle privacy but are not robust to a single drop-out.

Theorem 30. *There exists a protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ such that $(\mathcal{S} \circ \mathcal{R}^n)$ satisfies pure differential privacy but $(\mathcal{S} \circ \mathcal{R}^{n-1})$* does not *satisfy pure differential privacy.*

Proof. Define $\mathcal{R} : \{0,1\} \to \{1\}^*$ such that the length of the output (number of messages) is uniformly random over $\{0, \ldots, n+2\}$ on input 0 and uniformly random over $\{0, 1, n+1, n+2\}$ on input 1.

We first show that $(\mathcal{S} \circ \mathcal{R}^n)$ is ε-differentially private for a finite value of ε. This is achieved by arguing that, for every input $\vec{x}$, the length of $(\mathcal{S} \circ \mathcal{R}^n)(\vec{x})$ has support $G = \{0, \ldots, n^2 + 2n\}$. We use the notation $\mathrm{supp}(|(\mathcal{S} \circ \mathcal{R}^n)(\vec{x})|) = G$. This equivalence holds if and only if the two following statements are true: (i) the length of $(\mathcal{S} \circ \mathcal{R}^n)(\vec{x})$ must be some member of the set $G := \{0, \ldots, n^2 + 2n\}$ and (ii) each integer in G has a nonzero probability of being the length.

(i) is immediate from the specification of $\mathcal{R}$: the length is maximized when all users send $n+2$ messages and minimized when they send no messages. To prove (ii), we perform case analysis over $\vec{x}$.

When $\vec{x} = 0^n$, we shall use induction over the elements of G in order. The base case is immediate: $\mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^n)(0^n)| = 0\right] = \mathbb{P}\left[|\mathcal{R}(0)| = 0\right]^n > 0$. For the inductive step, we are given that $\mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^n)(0^n)| = g\right] > 0$ for some $g \in G - \{n^2 + 2n\}$ and we show that $\mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^n)(0^n)| = g + 1\right] > 0$. There must be a vector $\vec{g} \in \{0, \ldots, n+2\}^n$ such that $\sum g_i = g$ and $\prod_{j=1}^n \mathbb{P}\left[|\mathcal{R}(0)| = g_j\right] > 0$. Because $g < n^2 + 2n$, there must be some index i such that $g_i < n + 2$. Hence, define $\vec{g}'$ such that $g'_i = g_i + 1$ and $g'_j = g_j$ for all $j \neq i$. Now we have that $\mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^n)(0^n)| = g + 1\right] \geq \prod_{j=1}^n \mathbb{P}\left[|\mathcal{R}(0)| = g'_j\right] > 0$.

When $\vec{x} = 1^n$, the proof is similar except the inductive step proceeds via case analysis. If $0 \in \vec{g}$, we simply create $\vec{g}'$ by changing the 0 to 1. If $n+1 \in \vec{g}$ we create $\vec{g}'$ by changing the $n+1$ to $n+2$. Otherwise, there is some integer $k \geq 0$ such that $\vec{g}$ consists of k copies of $(n+2)$ and $n-k$ copies of 1. In this case, we construct $\vec{g}'$ which has $n-k-1$ copies of 0 and $k+1$ copies of $n+1$. In all cases, $\sum g'_j = 1 + \sum g_j$ and $\prod \mathbb{P}\left[\mathcal{R}(1) = g'_j\right] > 0$.

For any other choice of $\vec{x}$, the fact that $\text{supp}(|\mathcal{R}(1)|) \subset \text{supp}(|\mathcal{R}(0)|)$ implies

$$\text{supp}(|(\mathcal{S} \circ \mathcal{R}^n)(1^n)|) \subseteq \text{supp}(|(\mathcal{S} \circ \mathcal{R}^n)(\vec{x})|) \subseteq \text{supp}(|(\mathcal{S} \circ \mathcal{R}^n)(0^n)|)$$

so that all the supports are precisely G.

Now we show that $(\mathcal{S} \circ \mathcal{R}^{n-1})$ cannot satisfy pure differential privacy. Consider the neighboring inputs $\vec{x} := 0^{n-1}$ and $\vec{x}' := 0^{n-2}1$. There is a non-zero probability that $(\mathcal{S} \circ \mathcal{R}^{n-1})(\vec{x})$ has length n. However, this is impossible when the input is $\vec{x}'$, so the likelihood ratio is unbounded. □

Theorem 31. *There exists a protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ such that $(\mathcal{S} \circ \mathcal{R}^n)$ satisfies approximate differential privacy, but $(\mathcal{S} \circ \mathcal{R}^{n-1})$ does not satisfy* any *differential privacy.*

Proof. Define $\mathcal{R} : \{0,1\} \to \{1\}^*$ such that the length of the output is uniform over $\{0,1\}$ on input 0 and uniform over $\{n, n+1\}$ on input 1.

We first show that $(\mathcal{S} \circ \mathcal{R}^n)$ is (ε, δ)-differentially private for a finite value of ε and $\delta < 1$. This is achieved by arguing that, for any neighboring $\vec{x} \sim \vec{x}'$, the support of $(\mathcal{S} \circ \mathcal{R}^n)(\vec{x})$ intersects with that of $(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}')$. Let k be the number of times 0 occurs in $\vec{x}$; without loss of generality, assume that the number of times 0 occurs in $\vec{x}'$ is $k+1$. We have that

$$\begin{aligned}
&\mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^n)(\vec{x})| = n^2 - kn\right] \\
&\geq \mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^k)(0^k)| = 0\right] \cdot \mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^{n-k})(1^{n-k})| = (n-k) \cdot n\right] \\
&> 0
\end{aligned}$$

and that

$$\begin{aligned}
&\mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}')| = n^2 - kn\right] \\
&\geq \mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^k)(0^k)| = k\right] \cdot \mathbb{P}\left[|\mathcal{R}(0)| = 1\right] \cdot \mathbb{P}\left[|(\mathcal{S} \circ \mathcal{R}^{n-k-1})(1^{n-k-1})| = (n-k-1) \cdot (n+1)\right] \\
&> 0
\end{aligned}$$

Now we argue that $(\mathcal{S} \circ \mathcal{R}^{n-1})$ cannot satisfy any degree of differential privacy. Given $\vec{x} = 0^{n-1}$ and $\vec{x}' = 0^{n-2}, 1$, the maximum length of $(\mathcal{S} \circ \mathcal{R}^{n-1})(\vec{x})$ is $n-1$ while the minimum length of $(\mathcal{S} \circ \mathcal{R}^{n-1})(\vec{x}')$ is n. Hence, we have neighboring inputs but the supports of the induced distributions are disjoint. □