Differential Privacy in the Shuffle Model:
A Survey of Separations

Albert Cheu

May 14, 2022

1 Introduction

Many differentially private algorithms operate in the *central model*, also known as the trusted curator model. Here, a single analyzer has raw user data and its computations are insensitive to any one user's data point. But the fact that all users give their data to one party means that there is a single point of failure: the privacy of all users is contingent on the integrity of the analyzer.

There are a number of ways to model weaker trust in the analyzer. Perhaps the most well-known among them is the *local model*. Here, the dataset is a distributed object where each user holds a single element. To preserve their own privacy, each user randomizes their data point and submits the output to the analyzer. Because the signal from each user is hidden behind noise, there are a number of lower bounds on the error of locally private protocols that strongly separate the local model from the central model [15, 19, 12, 1, 42]. That is, the analyzer needs more samples (users) to achieve the same accuracy as in the central model. Locally private protocols are also more vulnerable to manipulation: by sending carefully distributed messages, malicious users can skew tests and estimates of distributions beyond simply changing the input of the protocol [24]. These negative results lead us to ask the following question:

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

Research into the *shuffle model* has given an answer to this question. Like the local model, users in a shuffle protocol produce messages by feeding their data into a local randomizer. 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.

This survey gives an overview of the recent surge of work in the shuffle model. We pay particular attention to results that characterize the strength of the model relative to the local and central models.

Outline. In Section 2, we establish the requisite privacy and model definitions. In Section 3, we contrast local model lower bounds with shuffle model upper bounds: there are problems for which additive error and sample complexity are much lower in the shuffle model. Then, in Section 4, we give techniques to show that the shuffle model (under natural constraints) is weaker than the central model. Finally, Section 5 gives a glimpse of what is possible in interactive variants of the model.

All these results focus on the accuracy of differentially private shuffle protocols. In Appendix A, we explore alternative models and compare them with the shuffle model. Appendix B contains an overview of shuffle protocols that are designed with the aim of reducing costs of transmission (e.g. number of messages and total number of bits consumed by messages). And Appendix C highlights two unusual shuffle protocols which pose a challenge to proving lower bounds.

Author's Note. Much of this survey is derived from the author's PhD. thesis. Notation and definition changes have been introduced to simplify the presentation

2 Preliminaries

We will use the notation $[d] = \{1, 2, \dots, d\}$, $\mathbb{N} = \{1, 2, \dots\}$. 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 [30]). 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 $T \subset \mathcal{Z}$,

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

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.

The *binomial mechanism* is a centrally private algorithm that has proven to be useful in the design and analysis of shuffle protocols.

Lemma 2 (Binomial Mechanism [29, 36]). *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 $\vec{x}, \vec{x}\,' \in \mathcal{X}^n$. There is a constant κ such that, for any $\ell \in \mathbb{N}$, $p \in (0,1)$, and $\varepsilon, \delta \in (0,1)$ satisfying*

$$\ell \cdot \min(p, 1-p) \geq \frac{\kappa}{\varepsilon^2} \cdot \log \frac{1}{\delta},$$

*the algorithm that samples $\eta \sim$ **Bin**(ℓ, p) and outputs $f(\vec{x}) + \eta$ is (ε, δ)-differentially private. The error is $O\left(\frac{1}{\varepsilon}\sqrt{\log \frac{1}{\delta}}\right)$ with constant probability.*

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 3 (Local Model [50, 33]). A protocol $\mathcal{P}$ in the *local model* consists of two randomized algorithms:

- A randomizer $\mathcal{R} : \mathcal{X} \times \{0,1\}^r \to \mathcal{Y}$ mapping a data point and public random bits to a message
- An analyzer $\mathcal{A} : \mathcal{Y}^n \times \{0,1\}^r \to \mathcal{Z}$ that computes on a vector of messages and public random bits

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

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

where W is a uniformly random member of $\{0,1\}^r$.

It is possible to extend the model definition to allow for multiple rounds of communication. To ease readability, we defer discussion of interactive protocols (local and shuffle) to a later section.

Suppose the privacy adversary wishes to target user i. In this model, the adversary's view is limited to the output of $\mathcal{R}(x_i, W)$ so we impose the privacy constraint on $\mathcal{R}$.

Definition 4 (DP in the Local Model [30, 43]). A protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is (ε, δ)-*local differentially private* if, for all $w \in \{0,1\}^r$, $\mathcal{R}(\cdot, w)$ is (ε, δ)-differentially private. That is, 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

To give intuition for the shuffle model, we start by sketching a preliminary version called the *single-message shuffle model*. Like the (one-round) local model, users execute $\mathcal{R}$ on their data to produce messages but users now trust a service 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. *Amplification-by-shuffling* lemmas quantify how well the privacy parameters are improved [32, 11, 34]. These lemmas provide a simple way to design protocols in the single-message shuffle model.

But the amplification lemmas do not apply to the relaxed version of the model where each user sends any (possibly randomized) number of messages to the shuffler. Here, we assume the shuffling prevents messages from the same sender from being linked with one another. We give a formal definition below:

Definition 5 (Shuffle Model [16, 25])**.** A protocol $\mathcal{P}$ in the *shuffle model* consists of three randomized algorithms:

- A *randomizer* $\mathcal{R} : \mathcal{X} \times \{0,1\}^r \to \mathcal{Y}^*$ mapping a data point and public random bits to (possibly variable-length) vectors. The length of the vector is the number of messages sent. If, on any input, 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* $\mathcal{A} : \mathcal{Y}^* \times \{0,1\}^r \to \mathcal{Z}$ that computes on a permutation of messages and public random bits.

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

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

where W is again the public random string. We assume that $\mathcal{R}$ and $\mathcal{A}$ have access to n.

Remark 6. *By making n accessible to the parties, we allow internal parameters to depend on n. This enables users to evenly distribute the responsibility of adding noise.*

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

Definition 7 (DP in the Shuffle Model [25])**.** A shuffle protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is (ε, δ)-*differentially private* if, for all $w \in \{0,1\}^r$ and all[1] $n \in \mathbb{N}$, the algorithm

$$(\mathcal{S} \circ \mathcal{R}^n)(\vec{x}) := \mathcal{S}(\mathcal{R}(x_1, w), \dots, \mathcal{R}(x_n, w))$$

is (ε, δ)-differentially private.

For brevity, we typically call these protocols "shuffle private." We will also drop the public randomness input if it is unused.

Note that Definition 7 assumes all users follow the protocol. Ideally, distributed protocols should still guarantee some level of privacy even when users are malicious. A simple attack is to drop out: let $\mathcal{S} \circ \mathcal{R}^{\gamma \cdot n}$ denote the case where a γ fraction of n users execute $\mathcal{R}$ but they are only given access to n (not γ). $\mathcal{S} \circ \mathcal{R}^{1 \cdot n}$ might satisfy a particular level of differential privacy but there could be a value $\gamma < 1$ where

[1] Some protocols assume lower bounds on n in order to invoke concentration arguments. These bounds will typically be small.

$\mathcal{S} \circ \mathcal{R}^{\gamma \cdot n}$ does not.[2] This motivates a definition of shuffle privacy that is *robust* to a malicious minority of users:

Definition 8 (Robust DP in the Shuffle Model)**.** A shuffle protocol is (ε, δ)-robustly differentially private if, for all $w \in \{0,1\}^r$, the algorithm

$$(\mathcal{S} \circ \mathcal{R}^{n/2})(\vec{x}) := \mathcal{S}(\mathcal{R}(x_1, w), \ldots, \mathcal{R}(x_{n/2}, w))$$

is (ε, δ)-differentially private.

Discussion of Definition. We have defined 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.

Also, observe that robustness is not immediately implied by the basic form of shuffle privacy in Definition 7. Appendix C describes protocols that satisfy shuffle privacy but are not robust to drop-outs.

Finally, the constant 1/2 in the definition (corresponding to the assumption of an honest majority) can be changed to an arbitrary constant without changing the asymptotics of the upper or lower bounds.

Comparison with prior definitions. We remark that the work by Balcer, Cheu, Joseph, and Mao [8]—which originally formalized robustness in the shuffle model—offers a definition where privacy parameters are functions of the unknown fraction of users who are honest. The functions should be continuous and non-increasing, meaning that the privacy guarantee gradually loosens from (ε, δ) to $(O(\varepsilon), O(\delta))$. Although more general, their definition demands more notation which we avoid for simplicity.

We also note that Definition 8 is essentially an adaptation of earlier work by Ács and Castelluccia [5] on distributed differential privacy.

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 $\delta = O(1/\text{poly}(n))$.

3.1 Binary Sums

In this problem, each user i has a bit $x_i \in \{0,1\}$ and the objective is to compute the sum. Dating back to Warner [50], *randomized response* is the canonical local protocol for this problem. The randomizer is below:

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

[2] 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 ensures that adding other messages does not affect this guarantee. Hence, it is without loss of generality to focus on drop-out attacks.

	Local	Shuffle
Error of	$\Omega\left(\frac{1}{\varepsilon}\sqrt{n}\right)$	$O\left(\frac{1}{\varepsilon}\right)$
Binary Sums	[15, 19]	[9, 27]
ℓ_∞-error of	$\Omega\left(\frac{1}{\varepsilon}\sqrt{n \log k}\right)$	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$
d-bin Histograms	[12]	[7]
Sample Complexity of	$\Omega\left(\frac{d}{\alpha^2\varepsilon^2}\right)$	$O\left(\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{d^{1/2}}{\alpha\varepsilon} + \frac{d^{1/2}}{\alpha^2}\right)$
α-Uniformity Testing	[1]	[27]
Sample Complexity of	$\Omega(\ell)$	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$
$(2,\ell)$-Pointer-Chasing	[42]	[7]

Table 1: Lower bounds in the local model aligned with upper bounds in the shuffle model

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 9 (Beimel et al. [15] & Chan et al. [19])**.** *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 $\alpha = \Omega(\frac{1}{\varepsilon}\sqrt{n})$.*

3.1.1 Shuffling Randomized Response

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. [25] 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 10 (Cheu et al. [25])**.** *There exists a choice of p such that the shuffle protocol $\mathcal{P}_{\mathsf{RR}} = (\mathcal{R}_{\mathsf{RR}}, \mathcal{A}_{\mathsf{RR}})$ is (ε,δ)-robustly private and computes binary sums up to additive error $O(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}})$ with constant probability.*

Proof. We will set p to a value $\Omega(\frac{1}{\varepsilon^2 n}\log\frac{1}{\delta})$. If this quantity exceeds $1/2$ (which occurs when n is not large enough), p must take a different form and the analysis will naturally change; we omit this technicality for neatness. Refer to [25] for more details.

Robust privacy: Assume without loss of generality that the honest majority is the set $[n/2]$. We leverage the fact that the view of an adversary is an unordered set of bits. This object contains as much information as the sum of those bits. More formally, given $\sum_{i=1}^{n/2} y_i$, the adversary can simulate a sample from $\mathcal{S}(y_1,\dots,y_{n/2})$: pick a uniformly random binary string of length $n/2$ and sum $\sum_{i=1}^{n/2} y_i$. This procedure is a post-processing operation, which means we only have to ensure the privacy of $\sum_{i=1}^{n/2} y_i$.

By construction, some set of users $H \subset [n/2]$ will report messages sampled from $\mathbf{Ber}(1/2)$ and the rest will report their true values. So for any fixed set H, $\sum_{i=1}^{n/2} y_i$ is a sample from $\sum_{i\in[n/2]-H} x_i + \mathbf{Bin}(|H|, 1/2)$. If we show $|H| \geq \frac{\kappa}{\varepsilon^2} \cdot \log\frac{1}{\delta}$ where κ is the constant in Lemma 2, then we can invoke the binomial mechanism to conclude that $\mathcal{S} \circ \mathcal{R}^{n/2}$ satisfies (ε, δ)-differential privacy.

Membership in H is a Bernoulli process, so $|H| \sim \mathbf{Bin}(n/2, p)$. Due to our choice of p, standard concentration arguments imply $|H| \geq \frac{\kappa}{\varepsilon^2} \cdot \log\frac{1}{\delta}$ with at least $1-\delta$ probability.

Accuracy: We bound the protocol's error under the assumption that all users are honest. 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 $O(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}})$. □

The Noise/Data Dichotomy. In randomized response, a message can either be a Bernoulli bit or a data bit. Balle, Bell, Gascón, and Nissim [11] prove that this dichotomy is in fact one instance of a general phenomenon: *any* locally private randomizer can be expressed as a mixture of noise and data. Specifically, there is a "blanket" distribution $\mathbf{B}$ and a parameter p such that, for any input x, the distribution of $\mathcal{R}(x)$ is equal to $p\mathbf{B} + (1-p)\mathbf{D}_x$ where $\mathbf{D}_x$ an input-dependent distribution (identity in the case of $\mathcal{R}_{\mathsf{RR}}$). [11] use this to prove their amplification-by-shuffling lemma. The work by Feldman et al. [34] strengthens the result by, roughly speaking, performing the noise/data decomposition on an input-by-input basis.

3.1.2 Other Protocols for Binary Sums

We remark that there are shuffle protocols which have properties not present in RR. These are achieved by leveraging the power of multiple messages. Table 2 presents their most salient features.

Error	No. Messages per User	Advantage over RR	Source
$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$	2	If sum is 0, estimate is 0 with prob. 1	[7]
$O\left(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}}\right)$	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right) *$	Symmetric noise	[8]
$O\left(\frac{1}{\varepsilon^{3/2}}\sqrt{\log\frac{1}{\varepsilon}}\right)$	$O\left(\frac{1}{\varepsilon}\log n\right)$	$\delta = 0$	[35]
$O\left(\frac{1}{\varepsilon}\right)$	$1 + O\left(\frac{1}{\varepsilon^2 n}\log^2\frac{1}{\delta}\right) *$	Optimal error	[37]
$O\left(\frac{1}{\varepsilon}\right)$	$\tilde{O}\left(\mathrm{poly}\left(n, \frac{1}{\varepsilon}\right)\right)$	$\delta = 0$ *and* optimal error	[27]

Table 2: Shuffle protocols for binary sums. Each message is one bit. "$*$" denotes a bound that holds in expectation over the randomness of all users.

We note that the protocol by Cheu & Yan [27] pays a large price in message complexity in order to achieve both optimal error and pure differential privacy. It is unclear if this price is necessary. Prior work by Ghazi, Golowich, Kumar, Manurangsi, Pagh, and Velingker[35] and Ghazi, Kumar, Manurangsi, and Pagh [37] were able to achieve only one of the two properties with much fewer messages.

3.2 Histograms

In this setting, each user has one value in the set $[d]$. Let c_j denote the count of j in the input dataset. The objective is to privately compute a vector $(\tilde{c}_1, \ldots, \tilde{c}_d)$ such that the ℓ_∞ distance from $(c_1, \ldots, c_d)$ is small. In other words, the output's maximum error should be low. This error must grow with d under local privacy:

Theorem 11 (Bassily & Smith [12]). *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 $\alpha = \Omega(\frac{1}{\varepsilon}\sqrt{n \log d})$*

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

Theorem 12 (Balcer et al. [7, 8]). *There is a shuffle protocol that satisfies (ε, δ)-robust differential privacy and outputs a histogram that 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 perform d different binary sums. A union bound suffices to upper bound the maximum magnitude of error. Basic composition ensures that we only pay a factor of 2 in the privacy parameters, since changing a user's value from x to x' only affects the counts of those two values.

In the shuffle model, the protocol executions can be done in parallel: each user simply labels their messages with the execution number. 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 tuples (j, a) and (j, b). To estimate c_j, we can feed the messages into the corresponding analyzer function $\mathcal{A}_j$.

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

We present the local randomizer of ZSUM below. r is a parameter to be determined.

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

Robust Privacy: As with RR, it suffices to prove privacy of the sum of the messages from honest users. But this quantity is exactly $\sum_{i=1}^{n/2} x_i + \eta$, where η is drawn from the distribution $\mathbf{Bin}(n/2, r)$. And by Lemma 2, it suffices to choose $r = 1 - \frac{\kappa}{\varepsilon^2 n} \cdot \log \frac{1}{\delta}$ for $(O(\varepsilon), \delta)$ privacy.[3]

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, $\mathbb{P}[\sum y_{i,1} + y_{i,2} \leq n] = 1$ 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 $\sum y_{i,1} + y_{i,2} > n$. □

In Appendix B, we give a more technically involved protocol that has improved communication complexity. We also summarize alternative histogram protocols

3.3 Uniformity testing

In *α-uniformity testing*, we assume each user has an i.i.d. sample from some probability distribution $\mathbf{D}$ over $[d]$. The objective is to report "uniform" with probability 2/3 when $\mathbf{D} = \mathbf{U}$ and "not uniform"

[3] If $n < \frac{2\kappa}{\varepsilon^2} \cdot \log \frac{1}{\delta}$, notice that $r > 1/2$. In this case, honest users can simply opt to report (0,0). Perfect privacy is achieved at the price of error $n = O(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta})$

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 d.

Theorem 13 (Acharya et al. [1]). *If an ε-locally private protocol performs α-uniformity testing, then its sample complexity is $\Omega(d/\alpha^2\varepsilon^2)$.*

But under robust shuffle privacy, it has been shown that the sample complexity has a leading term of $d^{2/3}$ instead of d. Balcer, Cheu, Joseph, and Mao [8] give the first protocol to achieve that bound. Canonne and Lyu [18] streamline its analysis and describe a *single-message* protocol of their own using privacy amplification. Both protocols demand approximate differential privacy.

The testing protocol by Cheu and Yan [27] attains pure differential privacy while maintaining the same leading $d^{2/3}$ term. It follows much the same "recipe" as that of Balcer et al. [8] which has two parts: a core testing protocol whose sample complexity scales with $d^{3/4}$ and a domain compression lemma that lets us reduce the sample complexity to one that scales with $d^{2/3}$. [27]'s core tester uses a pure DP counting protocol while [8] relies on an approx. DP counting protocol.

Theorem 14 (Cheu & Yan [27]). *There is a multi-message protocol that satisfies ε-robust shuffle privacy and solves α-uniformity testing with sample complexity*[4]

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

Proof. As previously mentioned, much of this construction is borrowed from Balcer et al. [8]. 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 $c_1,\ldots,c_d$ mutually independent, which simplifies the analysis.

Cai et al. [17] give a recipe for private uniformity testing under Poissonization. First, compute a private histogram $(\tilde{c}_1,\ldots,\tilde{c}_d)$. Then, compute the test statistic

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

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. [6] 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

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

Analysis in Acharya et al. [3] imply bounds on term Z in the two relevant cases: there is a constant t and a function $f(\alpha,m)$ such that

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

[4] Special thanks to Clément Canonne for simplifying the big-Oh expression.

If we 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

then combining with 1. and 2. implies that the value $t \cdot f(\alpha, m)$ serves as a threshold that successfully separates the two cases with constant probability.

Balcer et al. [8] describe a binary sum protocol which produces estimates with zero-mean symmetric noise.[5] A corollary is that there is a private histogram protocol where each η_j is an independent sample from zero-mean symmetric noise. (i) follows from the fact that the probability of $\eta_j > 0$ is $1/2$. (ii) follows from Chebyshev's inequality and the moments of η_j.

Cheu & Yan [27] follow precisely the same template except they deploy a binary sum protocol that satisfies pure differential privacy ($\delta = 0$) and $O(1/\varepsilon)$ error. □

We now sketch how to reduce the sample complexity dependence on d from $d^{3/4}$ to $d^{2/3}$. The technique is 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]) 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.

Lemma 15 (Domain Compression [2, 6])**.** *Let* $\mathbf{D}$ *be a distribution over* $[d]$*. For any partition* G *of* $[d]$ *into* $\hat{d} < d$ *groups* $G_1, \dots, G_{\hat{k}}$*, let* $\mathbf{D}_G$ *be the distribution over* $[\hat{k}]$ *with probability mass function* $\mathbb{P}\left[\mathbf{D}_G = \hat{j}\right] := \sum_{j \in G_{\hat{j}}} \mathbb{P}[\mathbf{D} = j]$*. If* G *is chosen uniformly at random, then with probability* $\geq 1/954$ *over* G*,*

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

Public randomness can be used to create the partition G. Users can then replace their data j with the partition $\hat{j}$ it belongs to. Running the protocol on the transformed dataset (with distance parameter $\hat{\alpha} := \alpha \frac{\sqrt{\hat{d}}}{477\sqrt{10d}}$) gives the final uniformity tester below:

Theorem 16 (Cheu & Yan [27])**.** *Fix any* $\varepsilon = O(1)$*, and* $0 < \alpha < 1$*. There exists a protocol that is* ε*-robustly shuffle private and solves* α*-uniformity testing with sample complexity*

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

3.4 Pointer-Chasing

The *pointer chasing problem* is denoted $\mathrm{PC}(k, \ell)$ where $k, \ell \in \mathbb{N}$. 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* $\mathrm{PC}(k, \ell)$ *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 k-th integer in the sequence $a_1, b_{a_1}, a_{b_{a_1}} \dots$ with constant probability.

Joseph, Mao, and Roth show that the sample complexity of $\mathrm{PC}(2, \ell)$ under local privacy must scale at least linearly with ℓ.

[5] At a high level, each user sends a random number of $\mathbf{Ber}(1/2)$ messages. The aggregate number of such bits is guaranteed to be $\Theta(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ with $1 - O(\delta)$ probability.

Theorem 17 (Joseph et al. [42])**.** *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 [7])**.** *There is a $8\cdot(\ell!)^2$-message protocol that satisfies (ε,δ)-robust shuffle privacy 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 $\geq t+1$ times with constant probability. Taking universe size $d=4\cdot(\ell!)^2$, we then use the histogram protocol built atop ZSUM (Theorem 12). 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 known separations between the (one-round) shuffle model and the central model. The proofs thus far require some natural structural constraint.

4.1 Single-message Shuffle Privacy

The first class of lower bounds hold for protocols wherein each user sends exactly one message with probability 1.[6] We begin with a negative result for bounded-value sums proved by Balle, Bell, Gascón, and Nissim [11].

Theorem 19 (Balle et al. [11])**.** *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 contrast, the centrally private Laplace mechanism achieves mean-squared error of $O(1/\varepsilon^2)$.

The techniques used to prove the above are specific to bounded-value sums. A more general technique is to study what happens when we remove the shuffler from a single-message protocol. This takes us to what we can call *removal lemmas*

Lemma 20 (Balcer & Cheu [7])**.** *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 the shuffle protocol as a whole is ε-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 removal lemma accommodates the relaxation.

Lemma 21 (Cheu et al. [25])**.** *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.*

[6] The lower bounds also hold in the case where users send *at most* one message. This is proven by a simple transformation: send a dummy symbol $\perp$ to denote the no-message event.

Thus, we can invoke any local model lower bound that holds for $(\varepsilon + \ln n, \delta)$ privacy. As an example, the recipe implies the following lower bound on the error of histograms.

Theorem 22 (Ghazi et al. [36])**.** *Any single-message protocol that satisfies $(1, o(1/n))$-shuffle privacy and estimates a d-bin histogram with ℓ_∞ error $n/10$ must have $n = \Omega(\frac{\log d}{\log\log d})$.*

In contrast, there is a central model algorithm where $n = O(1)$ suffices for the same privacy and accuracy regimes.

4.2 m-message Shuffle Privacy

A natural idea is to somehow extend the removal lemma from the single-message case to the m-message case. But there are differentially private shuffle protocols whose randomizers are *not* differentially private. For example, an adversary can recover the input of $\mathcal{R}_{\mathsf{ZSUM}}$ by simply looking at the first bit of the output. Other examples can be found in Appendix C.

Despite this hurdle, two works manage to prove lower bounds for m-message protocols. These lower bounds make the simplifying assumption that the local randomizer sorts (or shuffles) its output messages before giving them to the shuffler. This does not affect accuracy or privacy because the local sorting (or local shuffling) is undone by the shuffler anyway.

4.2.1 Approach 1

One paper by Beimel, Haitner, Nissim, and Stemmer [14] obtains a bound on the mutual information between the output of an m-message randomizer and uniformly random input.

Lemma 23. *Let $\mathcal{P} = (\mathcal{R}, \mathcal{A})$[7] be an m-message (ε, δ)-shuffle private protocol and let $Z_1, \ldots, Z_n \in \mathcal{X}$ be (possibly correlated) random variables. In the execution of $\mathcal{P}$ on input $Z_1, \ldots, Z_n$, let Y_i be the (sorted) output of the i-th user and let W denote the public randomness. For any $i \in [n]$, if Z_i is uniformly random over $\mathcal{X}$, then*

$$I(Y_i, W; Z_i) = O\left((en)^m \cdot \left(\varepsilon^2 + \frac{\delta}{\varepsilon}\log|\mathcal{X}| + \frac{\delta}{\varepsilon}\log\frac{\varepsilon}{\delta}\right) + m\log n\right).$$

Proof Sketch. Given (ε, δ)-shuffle private protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$, we can create a (ε, δ)-locally private randomizer $\mathcal{R}_{\mathcal{P}}$: on input x, W, obtain nm messages by executing $(\mathcal{S} \circ \mathcal{R}^n)(U_1, U_2, \ldots, U_{n-1}, x)$ where U_i is uniformly random, and then output a random (sorted) subset of m messages. Privacy follows from post-processing.

Now let $Y_i' \leftarrow \mathcal{R}_{\mathcal{P}}(Z_i, W)$. Prior work has shown that $I(Y_i', W; Z_i) = O(\varepsilon^2 + \frac{\delta}{\varepsilon}\log|\mathcal{X}| + \frac{\delta}{\varepsilon}\log\frac{\varepsilon}{\delta})$. Then we use the fact that Y_i' coincides with Y_i with probability $\binom{nm}{m}^{-1}$. □

The above lemma is then used to obtain a lower bound for the *common element* problem. Refer to [14] for the full details.

4.2.2 Approach 2

A paper by Chen, Ghazi, Kumar, and Manurangsi [20] takes a different approach. They define a relaxation of differentially private algorithms—called *dominated algorithms*— and then argue that the local randomizer of a shuffle private protocol satisfies that definition.

Definition 24 (Chen et al. [20])**.** An algorithm $\mathcal{R} : \mathcal{X} \times \{0,1\}^* \to \mathcal{Y}$ is (ε, δ)-dominated if there exists a distribution $\mathbf{D}$ such that for all $x \in \mathcal{X}$, all $w \in \{0,1\}^r$, and all $Y \in \mathcal{Y}$, $\mathbb{P}[\mathcal{R}(x,w) \in Y] \le e^\varepsilon \cdot \mathbb{P}[\mathbf{D} \in Y] + \delta$

[7] The original statement allows for different users to run different randomizers, but we omit that degree of freedom for simplicity

Notice that the above definition is a one-sided variant of differential privacy since it does not require the probability mass function of $\mathcal{R}(x,w)$ to dominate that of $\mathbf{D}$.

Lemma 25. *If $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ is an m-message (ε, δ)-shuffle private protocol, then $\mathcal{R}$ is $(\varepsilon + m\ln(en), \delta)$-dominated.*

Using the above, Chen et al. derive a lower bound for parity learning:

Theorem 26. *If $\mathcal{P}$ is a m-message shuffle protocol that solves d-dimensional parity learning, then its sample complexity is $\Omega(2^{d/(m+1)})$.*

In contrast, Kasiviswanathan, Lee, Nissim, Raskhodnikova, and Smith [43] show that centrally private parity learning is possible with just $O(d)$ samples.

4.3 Robust Shuffle Privacy

The third class of lower bound applies to robustly shuffle private protocols. To obtain these results, we again develop reductions, but this time to the online model. Briefly, an online algorithm receives user data one at a time and updates its internal state upon reading each input. The algorithm produces output when the stream ends.

How do we define privacy in the online model? Dwork, Naor, Pitassi, Rothblum, and Yekhanin [31] propose *pan-privacy*: for any time t, the joint distribution of the internal state at time t and the output should be differentially private. This models one-time violations of the algorithm's integrity (i.e. a hack, a subpoena, or a change in ownership).

Balcer, Cheu, Joseph, and Mao [8] describe a generic transformation from robust shuffle privacy to pan-privacy that preserves accuracy for many statistical problems. Thus, existing lower bounds that hold under pan-privacy—for the distinct elements and uniformity testing problems—carry over to robust shuffle privacy. Cheu & Ullman [26] and Nissim & Yan [45] obtain new lower bounds for pan-private selection and parity learning, which again imply lower bounds for robust shuffle privacy. This second batch of results imply exponentially large separations in sample complexity between robust shuffle privacy and central privacy. Refer to Table 3 for an overview of these results.

In the thesis by Cheu [22], the recipe is somewhat simplified. The key observation is that lower bounds for pan-privacy typically only require the privacy of the internal state and not that of the state-output pair. [22] uses *internal privacy* to refer to this weaker notion. Transforming robustly shuffle private protocols to internally private algorithms is a little easier than transforming them to pan-private algorithms, while still producing the same results.[8]

In the following lemma, $\mathbf{U}$ is any distribution[9] over the data universe $\mathcal{X}$ and let $\mathbf{U}^n$ be the corresponding product distribution over $\mathcal{X}^n$. For any other distribution $\mathbf{D}$, let $\mathbf{D}_{(p)}$ be the mixture $p \cdot \mathbf{D} + (1-p) \cdot \mathbf{U}$.

Lemma 27 (Balcer et al. [8], Cheu [22])**.** *Let $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ be an (ε, δ)-robustly shuffle private protocol. There is an (ε, δ)-internally private algorithm $\mathcal{Q}_{\mathcal{P}}$ such that*

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

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

$$d_{\mathrm{TV}}(\mathcal{Q}_{\mathcal{P}}(\mathbf{D}^{n/2}), \mathcal{P}(\mathbf{D}^n_{(1/4)})) < 1/6. \tag{3}$$

Proof Sketch. The online algorithm's initial internal state will be the output of $(\mathcal{S} \circ \mathcal{R}^{n/2})$ run on $n/2$ i.i.d. samples from $\mathbf{U}$. Each time a user's data point is read, the algorithm will execute $\mathcal{R}$ on it and add the

[8] It also avoids a technical limitation of the original transformation, which is that privacy parameter needs to be known when a third of users participate.

[9] As the symbol suggests, it is typically the uniform distribution

Table 3: Comparison of impossibility results for robust shuffle privacy with centrally private algorithms. d and α are dimension and error parameters, respectively. k is the number of inputs to the learned parity function. For simplicity, we use $\varepsilon = \hat{\varepsilon}(1/2)$ and $\delta = \hat{\delta}(1/2)$. $*$ indicates that $\delta \log(\binom{d}{\leq k}/\delta) \ll \alpha^2\varepsilon^2/\binom{d}{\leq k}$.

		Robust Shuffle Privacy	Central Privacy
Additive Error of	Distinct Elements	$\Omega\left(\sqrt{\frac{d}{\varepsilon}} + \frac{1}{\varepsilon}\right)$ [8] $(n \geq 2d)$	$O\left(\frac{1}{\varepsilon}\right)$ [30]
Sample Complexity of	Uniformity Testing	$\Omega\left(\frac{d^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{\sqrt{d}}{\alpha^2} + \frac{1}{\alpha\varepsilon}\right)$ [8] $(\delta = 0)$	$O\left(\frac{\sqrt{d}}{\alpha^2} + \frac{\sqrt{d}}{\alpha\varepsilon} + \frac{d^{1/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{1}{\alpha\varepsilon}\right)$ [4]
	Parity Learning	$\Omega\left(\sqrt{\binom{d}{\leq k}}/\alpha\varepsilon\right)$ [26] agnostic, [45] realizable $*$	$O(\log \binom{d}{\leq k})$ [43]

messages to the internal state (inserted in some random position). This ensures internal privacy because any internal state is equivalent to the output of the shuffler when the protocol is run on (at least) $n/2$ data points.

The output of $\mathcal{Q}_{\mathcal{P}}$ is simply the execution of $\mathcal{A}$ on the final state. (2) is immediate from the construction. To obtain (3), we begin with the observation that the final internal state consists of messages produced by running the protocol on independent samples from $\mathbf{U}, \ldots, \mathbf{U}, \mathbf{D}, \ldots, \mathbf{D}$. This looks almost like $\mathbf{D}_{(1/2)}$ except that the number of samples from $\mathbf{D}$ should be binomial. We correct this by slightly modifying the transformation: replace the first $\mathbf{Bin}(n/2, q)$ user data with samples from $\mathbf{U}$. q is chosen so that the shuffled set of samples approximates samples from $\mathbf{D}_{(1/4)}$. The modification does not invalidate our preceding arguments. □

5 The Promise of Interactivity

Thus far, we have limited our attention to one-round shuffle protocols. We shall now explore what shuffle protocols can do with multiple rounds of communication and how they stack up against centrally private algorithms.

5.1 Sequential Interactivity (S.I.)

To start, it will help to understand sequentially interactive local protocols. Here, each user sends only one message but the randomizer of user i can depend on the *transcript* generated by users $1, \ldots, i-1$. This is useful when implementing private iterative methods like gradient descent. Strong separations are known to exist between one-round and sequentially interactive local privacy. Joseph, Mao, and Roth [42] show that two rounds suffice to solve pointer-chasing $\mathrm{PC}(2, \ell)$ with sample complexity $O_\varepsilon(\log \ell)$. This is exponentially smaller than the lower bound of $\Omega_\varepsilon(\ell)$ in the one-round case (Theorem 17).

Given that S.I. provably enhances the local model, how can we adapt it to the shuffle model?

Approach 1. One option is to re-interpret the shuffler as an anonymity service: users are shuffled u.a.r. and then the analyzer deploys a sequentially interactive local protocol.[10] The recent amplification-by-

[10] An equivalent interpretation is that, at the beginning of each round of the S.I. local protocol, a middle-man samples a random user without replacement.

Table 4: Comparison of positive results in the S.I. shuffle model with central model counterparts. For brevity, we suppress the term $\sum_{a\in[k]:\Delta_a>0}\frac{\log T}{\Delta_a}$ present in both MAB bounds and the term $1/\sqrt{n}$ in the SCO bounds. SCO bounds also omit logarithmic factors, as well as convexity and smoothness parameters.

		S.I. Shuffle Privacy	Central Privacy
Regret of	k-arm	$O\left(\frac{k}{\varepsilon}\sqrt{\log\frac{1}{\delta}}\log T\right)$	$O(k/\varepsilon)$
	bandit	[48]	[49]
	Convex, Non-Smooth	$O(d^{1/3}/\varepsilon^{2/3}n^{2/3})$	
SCO	Convex, Smooth	$O(d^{2/5}/\varepsilon^{4/5}n^{4/5})$	$O(\sqrt{d}/\varepsilon n)$
error	Strongly Convex, Non-Smooth	$O(d^{2/3}/\varepsilon^{4/3}n^{4/3})$	[13]
	Strongly Convex, Smooth	$O(d/\varepsilon^2 n^2)$	

shuffling lemma by Feldman, McMillan, and Talwar [34] holds in this version of the model. Notice that if the randomizer does not get updated over time, we are just running a one-round single-message shuffle protocol. Also observe that it is not possible to run multi-message protocols in this variant of the shuffle model, since the random permutation is limited to the users and not the messages.

Approach 2. An alternative way to adapt S.I. is to simply run one-round shuffle protocols on disjoint batches of users. The i-th protocol can depend on the transcript from protocols $1,\dots,i-1$ and can be multi-message. Summarized in Table 4, two recent works have described protocols in this model. Tenenbaum, Kaplan, Mansour, Stemmer [48] study the multi-arm bandit problem. The authors give cumulative regret bounds that match those of the central model up to logarithmic factors. Cheu, Joseph, Mao, and Peng [23] focus instead on the problem of stochastic convex optimization (SCO). They describe a one-round vector summation protocol that is repeatedly called inside gradient descent algorithms.

5.2 Full Interactivity (F.I.)

In fully interactive local protocols, a user can communicate with the analyzer multiple times. The transcript of all the user's messages must be differentially private.

We can adapt F.I. to the shuffle model in the following way: run one-round shuffle protocols on batches of users that are not necessarily disjoint. The transcript of a fully interactive shuffle protocol is the entire list of the outputs of the shuffler. As with local protocols, this transcript must be differentially private. As an example, Cheu et al. [23] give a SCO protocol that relies on this ability to query a user multiple times.

Beimel et al. [14] describe a very powerful transformation that shows fully interactive shuffle private protocols can be as powerful as centrally private ones(!)

Theorem 28. *Let $\mathcal{M}$ be an arbitrary (central model) randomized algorithm. Assuming an honest majority and semi-honest corruptions, there exists a two-round fully interactive shuffle protocol $\mathcal{P}_{\mathcal{M}}$ that simulates $\mathcal{M}$.*

Proof Sketch. The idea is to simulate an information-theoretically secure multi-party computation protocol by Applebaum, Brakersky, and Tsabary (ABT), the source of the honest majority requirement. The MPC protocol relies on secure channels of communication; to simulate these channels in the shuffle model, Beimel et al. use one-time pads.

We begin with a simple building block: Alice and Bob want to agree on one random bit, with one party designated as "leader." As usual, the adversary's view is limited to the output of the shuffler.

Suppose both Alice and Bob each flip one fair coin and send their bits. By examining the output of the shuffler, each party can learn what the leader sampled.[11] However, if both have 0 or both have 1, the adversary learns both their bits. This has a 1/2 chance of occurring, so they repeat the process enough times to drive the probability down. Note that these repetitions can be done in parallel by labeling each bit with a repetition number. When there are $n > 2$ users, we label each message with the pair of users who will read them.

Naively combining the above key agreement with ABT leads to a three-round protocol (one for key agreement and two for ABT). Beimel et al. show how to use the leftover hash lemma to send a message and the pad at the same time, reducing the number of rounds to two. □

6 Open Questions

How can we close the gap between the amplification and removal lemmas? The best-known amplification lemma has constraints on the number of users and privacy parameters. In particular, they do not apply to $(\ln en, \delta)$-private randomizers. Randomizers with these parameters are created by the removal lemma by Cheu et al (Lemma 21). If the removal lemma guarantees could be tightened (or the amplification constraints could be loosened), then we would be able to show that a single-message shuffle protocol is intrinsically robust: when only a constant fraction γ of users participate, amplification of the randomizer's privacy guarantee would give us a concrete privacy parameter for $\mathcal{S} \circ \mathcal{R}^{\gamma n}$.

What is the optimal sample complexity of uniformity testing under approximate differential privacy? We have matching upper and lower bounds for testing under pure differential privacy. Prior work has shown that, in the central model, pure d.p. is not a stronger constraint on the sample complexity of a binary decision problem than approximate d.p. But is this the case for pan-privacy? Robust shuffle privacy?

What are the limits of S.I. protocols? It appears difficult to perform the same level of simulation as done in the fully interactive setting. There may be a way to adapt the strong lower bounds developed by Joseph et al. [42, 41]. Note that we can ask this question for both approaches of defining S.I. shuffle protocols.

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, 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*

[11] We can use the analyzer as a referee to relay the shuffler's output. Alternatively, we could model the shuffler as having the ability to broadcast its output (as done by Beimel et al.).

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. *CoRR*, abs/1911.06879, 2019.

[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] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 464–473. IEEE Computer Society, 2014.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] Clément L. Canonne and Hongyi Lyu. Uniformity testing in the shuffle model: Simpler, better, faster. In Karl Bringmann and Timothy Chan, editors, *5th Symposium on Simplicity in Algorithms, SOSA@SODA 2022, Virtual Conference, January 10-11, 2022*, pages 182–202. SIAM, 2022.

[19] 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.

[20] 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.

[21] Wei-Ning Chen, Christopher A. Choquette-Choo, Peter Kairouz, and Ananda Theertha Suresh. The fundamental price of secure aggregation in differentially private federated learning. *CoRR*, abs/2203.03761, 2022.

[22] Albert Cheu. *Differential Privacy in the Shuffle Model*. PhD thesis, Khoury College of Computer Sciences, Northeastern University, 2021. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-05-29.

[23] Albert Cheu, Matthew Joseph, Jieming Mao, and Binghui Peng. Shuffle private stochastic convex optimization. *CoRR*, abs/2106.09805, 2021.

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

[25] 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.

[26] Albert Cheu and Jonathan R. Ullman. The limits of pan privacy and shuffle privacy for learning and estimation. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1081–1094. ACM, 2021.

[27] Albert Cheu and Chao Yan. Pure differential privacy from secure intermediaries. *CoRR*, abs/2112.10032, 2021.

[28] Albert Cheu and Maxim Zhilyaev. Differentially private histograms in the shuffle model from fake users. *CoRR*, abs/2104.02739, 2021.

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Rasmus Pagh. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3505–3514. PMLR, 2020.

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

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil P. Vadhan. The limits of two-party differential privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 81–90. IEEE Computer Society, 2010.

[45] Kobbi Nissim and Chao Yan. The sample complexity of distribution-free parity learning in the robust shuffle model. *CoRR*, abs/2103.15690, 2021.

[46] Thomas Steinke. Multi-central differential privacy. *CoRR*, abs/2009.05401, 2020.

[47] Kunal Talwar. Differential secrecy for distributed data and applications to robust differentially secure vector summation. *CoRR*, abs/2202.10618, 2022.

[48] Jay Tenenbaum, Haim Kaplan, Yishay Mansour, and Uri Stemmer. Differentially private multi-armed bandits in the shuffle model. *CoRR*, abs/2106.02900, 2021.

[49] Aristide C. Y. Tossou and Christos Dimitrakakis. Algorithms for differentially private multi-armed bandits. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2087–2093. AAAI Press, 2016.

[50] 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 Other Models of Distributed Differential Privacy

Here, we discuss other distributed models and compare them with the shuffle model.

Secure Aggregation Model. In this model, there is a trusted service or functionality called the aggregator. Much like the shuffle model, users send their messages to the aggregator who then reports a value to the analyzer. The aggregator's output is the sum of user messages, modulo some modulus. [12]

Ishai, Kushilevitz, Ostrovsky, and Sahai [40] show that we can perform secure aggregation the shuffle model: each user samples a set of m values uniformly over sets that add up to their sensitive value. The protocol guarantees that, for large enough m and any inputs $\vec{x}, \vec{x}'$ that have the same sum, the distribution of the multiset of all user shares does not change significantly between $\vec{x}, \vec{x}'$. We will not provide the security proof for space, but we will use the construction in Appendix B.

Conversely, it is not difficult to show that a protocol in the shuffle model implies one in the secure aggregation model. A user could encode the set of m messages they intend to send as an integer where the j-th digit is the count of j in that message set. Adding up these encodings will yield the histogram of all user messages, which contains exactly the same information as the shuffled set of them.

Multi-central Model. In the work by Steinke [46], we find a model where a user can communicate with any subset of $k > 1$ servers but they are only guaranteed at least one honest server. The honest server(s) must communicate in a manner such that the view of the dishonest servers is insensitive to any single user's contribution.

Steinke shows that secret sharing can be combined with server-side noise to compute differentially private sums. One limitation of this protocol is that a malicious user can influence the sum by a magnitude

[12] In the language of Cheu and Yan [27], the shuffler and the aggregator are two realizations of a *secure intermediary*.

as large as the modulus of secret sharing, which needs to be at least as large as n. Recently, Talwar [47] offers an alternative protocol which verifies the magnitude of the user's contribution before adding it.

Multi-central protocols are at least as powerful as shuffle protocols with finite communication complexity. This follows from (1) the earlier observation that we can losslessly transform a multi-set of messages into a (large) integer and (2) secret sharing for summation.

Instead of that transformation-based approach, Steinke argues that we can *directly execute* shuffle protocols using public key cryptography (onion encryption). Furthermore, differentially private selection can be performed using $\log d$ samples when given access to an MPC implementation of argmax. In contrast, the techniques used in [26] imply selection demands d samples under robust shuffle privacy.

Two-Party Protocols. The work by McGregor, Mironov, Pitassi, Reingold, Talwar, and Vadhan [44] formalizes the following scenario: there are two servers, two disjoint datasets (no user appears in both), and server $j \in \{0,1\}$ has exclusive access to dataset j. The servers communicate with one another across rounds. An honest server j should interact with (potentially malicious) server $1-j$ such that the transcript is simulatable by a differentially private algorithm on dataset j.

This model is very close to the central model, since any "i.i.d.-style" problem like mean estimation, uniformity testing, and learning can simply be solved by each server on their own. Indeed, it is not hard to see that protocols in the distributed models we have seen so far can be simulated by two-party protocols. Still, McGregor et al. derive a lower bound on the *inner product* problem, where the goal is to estimate the inner product between the two datasets. $O(1/\varepsilon)$ error is possible in the central model via the Laplace mechansim but $\Omega(\sqrt{n})$ is necessary for the two-party model (and the others that can be simulated by it)

B Message Complexity and Communication Complexity

In any real implementation of a shuffle protocol, users will have to transmit their messages across a network. The two critical metrics are the number of messages sent by each user and the total number of bits they consume. We use *message complexity* to refer to the former and *communication complexity* to refer to the latter. In this Appendix, we give an overview of protocols that are designed to minimize one or both of these quantities. We also take a glimpse at lower bounds.

B.1 Communication-efficient Bounded-value Sums

In this setting, users have values in the interval $[0,1]$ and the objective is to privately compute their sum. It is possible to use a binary sum protocol for this problem: a fixed-point representation can transform a continuous value into a set of zeroes and ones, upon which we perform the local ranomization. A longer fixed-point representation reduces the rounding error, but increases the noise needed for privacy; two works [25, 23] show that $\sqrt{n}$-long representation suffices.

The downside of the above approach is that the message complexity —and thus the communication complexity— scales with $\sqrt{n}$. To rectify this, Balle et al. [9, 10] and Ghazi et al. [39, 38] use a different reduction that leads to a logarithmic communication complexity.

Theorem 29. *There is an (ε,δ)-shuffle private protocol for bounded-value sums with error $O(\frac{1}{\varepsilon})$ where each user sends $1+O(\frac{\log(1/\delta)}{\log(n)})$ messages, each consisting of $O(\log n)$ bits.*

Proof Sketch. At a high level, the goal is to simulate the symmetric geometric distribution $\mathbf{SG}(\varepsilon)$, also known as the discrete Laplace distribution. In the central model, adding such noise suffices for pure differential privacy.

The first step is to equate a sample from the $\mathbf{SG}(\varepsilon)$ with the sum of n samples from another distribution $\mathbf{D}_\varepsilon$. This property is called *infinite divisibility*, most obvious in the Gaussian distribution. The next step is to recall the modular arithmetic (or secure aggregation) protocol $\mathcal{P}_{\mathsf{MOD}} = (\mathcal{R}_{\mathsf{MOD}}, \mathcal{A}_{\mathsf{MOD}})$ by Ishai et al. [40]. It ensures that two input datasets with the same sum (modulo some modulus) cause the protocol to produce a shuffled set of messages that are δ-close in statistical distance. Finally, we 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. Due to our use of MOD, the output of the shuffler $(\mathcal{S} \circ \mathcal{R}^n)(x_1, \dots, x_n)$ is δ-close to the output of the algorithm $(\mathcal{S} \circ \mathcal{R}^n_{\mathsf{MOD}})(\sum y_i, 0, \dots, 0)$. This closeness suffices for approximate differential privacy. The error bound $O(\frac{1}{\varepsilon})$ because we are simulating the geometric mechanism and sums exceed the modulus with very low probability (assuming the modulus is large).

Refer to Balle et al. [10] and Ghazi et al. [38] for analyses of the message complexity of MOD. □

As an aside, Cheu and Yan [27] follow much the same template, except that the security property of their MOD replaces statistical distance with one derived from the definition of pure differential privacy. But this variant protocol demands exponentially more bits.

B.2 Almost-communication-efficient Histograms with Domain-Independent Error

Here, we describe a protocol that has the same asymptotic error as the protocol by Balcer & Cheu but reduced communication complexity.

Theorem 30. *Fix any $T \in \mathbb{N}$ and privacy parameters $0 < \varepsilon, \delta = O(1)$. There exists an (ε, δ)-private shuffle protocol which estimates histograms up to ℓ_∞ error $O(T^2 \log(T/\delta)/\varepsilon^2)$ with at least $99/100 - \delta$ probability and consumes $O\left(\frac{T^3 d^{1/T}}{\varepsilon^2} \log \frac{T}{\delta}\right)$ messages of length $O\left(\log Tn + \frac{1}{T} \log d\right)$.*

The construction proceeds in two steps. We first make an inefficient but accurate protocol, then describe a technique to reduce its communication complexity.

B.2.1 An opt-in protocol

Derived from conversations with Maxim Zhilyaev, this protocol reports private histograms such that the ℓ_∞ error is $O(\log(1/\delta)/\varepsilon^2)$ with $1-\delta$ probability. At a high level, each user "opts-in" to contributing noise to the count of each universe element. Much like in the analysis of shuffled randomized response, the size of this opt-in set only depends on the privacy parameters (and not n or d). This will in fact determine the maximum error of the histogram.

The first message user i sends is their true value x_i. Their second message is a bit b_i drawn from $\mathbf{Ber}(p)$ where $p = \Theta(\log(1/\delta)/\varepsilon^2 n)$. This bit determines whether or not the user opts-in: if $b_i = 1$, they will also flip d fair coins. If the j-th coin is heads, then they send j as yet another message.

To prove this protocol is differentially private, let H be the set of all users i where $b_i = 1$. We leverage the following concentration result: for sufficiently large n, $|H| \geq \frac{\kappa}{\varepsilon^2} \log \frac{1}{\delta}$ with probability $\geq 1-\delta$ where κ is the constant from Lemma 2. This implies differential privacy: the noise in the frequency of each j is an independent sample from $\mathbf{Bin}(|H|, 1/2)$ and $|H|$ is sufficiently large to ensure that additive noise offers (ε, δ)-privacy.

Error is also low. Because the noise on any bin is drawn from $\mathbf{Bin}(|H|, 1/2)$, we have that the ℓ_∞ error is $\leq |H| = O(\log(1/\delta)/\varepsilon^2)$ with probability $\geq 1-\delta$. Note that the analyzer can compute $|H|$ by simply adding up the one-bit messages. Also, we can derive the "zero-maps-to-zero" property from Balcer-Cheu by truncating estimates to 0 if they are at most $|H|$.

What is the communication complexity? Each user sends a one-bit message alongside at least one $\log_2 d$-bit message. The number of $\log_2 d$-bit messages sent by a user is a random variable with expectation

$1 + p \cdot (d/2) = 1 + \Theta(d \log(1/\delta)/\varepsilon^2 n)$. Contrast this with $1 + \Theta(d \cdot (1 - \log(1/\delta)/\varepsilon^2 n))$ in the protocol by Balcer & Cheu.

B.2.2 Count-Min Template

This meta-protocol samples random hash functions and repeatedly executes a subroutine for computing histograms on hashed data. The number of repetitions determines both the privacy parameters and the size of the hashed domain. See pseudocode in Algorithms 1 and 2

```
Algorithm 1: $\mathcal{R}_{CM}$ a local randomizer for histograms
Input: $x \in [d]$; parameters $T, \hat{d} \in \mathbb{N}$; randomizer $\mathcal{R} : [\hat{d}] \to \mathcal{Y}^*$
Output: $\vec{y} \in ([T] \times \mathcal{Y})^*$
Obtain hash functions $\{h^{(t)} : [d] \to [\hat{d}]\}$ from public randomness.
Initialize $\vec{y} \leftarrow \emptyset$
For $t \in [T]$
    Compute $\vec{y}^{(t)} \leftarrow \mathcal{R}(h^{(t)}(x))$
    For $y \in \vec{y}^{(t)}$
        Append $(t, y)$ to $\vec{y}$
Return $\vec{y}$
```

```
Algorithm 2: $\mathcal{A}_{CM}$ an analyzer for histograms
Input: $\vec{y} \in ([T] \times \mathcal{Y})^*$; parameters $T, \hat{d} \in \mathbb{N}$; analyzer $\mathcal{A} : \mathcal{Y}^* \to \mathbb{R}^{\hat{d}}$
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 $\hat{z}^{(t)} \leftarrow \mathcal{A}(\vec{y}^{(t)})$
    For $j \in [d]$
        $\hat{j} \leftarrow h^{(t)}(j)$
        $z_j \leftarrow \min(z_j, \hat{z}^{(t)}_{\hat{j}})$
Return $\vec{z}$
```

Theorem 31. *Fix any number of users n, domain size d and natural number T. Let $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ be any shuffle protocol for computing d-bin histograms where (1) each user sends, in expectation, $M(d)$ messages of length $L(d)$ (2) (ε, δ)-privacy is offered to any user and (3) with probability $\geq 1 - \beta$, the ℓ_∞ error is $\alpha(d, \beta)$. If we instantiate $\mathcal{P}_{CM} = (\mathcal{R}_{CM}, \mathcal{A}_{CM})$ with that $\mathcal{P}$ and parameters $T, \hat{d} \leftarrow \lceil n \cdot (100d)^{1/T} \rceil$, then*

1. *each user sends, in expectation, $T \cdot M(\hat{d})$ messages of length $L(\hat{d}) + \log T$*
2. *$\mathcal{P}_{CM}$ is $(T\varepsilon, T\delta)$-private*
3. *with probability $\geq 99/100 - \beta$, the ℓ_∞ error is $\alpha(\hat{d}, \beta/T)$.*

Proof. Since the randomizer executes $\mathcal{R}$ exactly T times on hashed user data, labeling messages with the execution number each time, Item 1 is immediate from substitution and Item 2 follows from basic composition.

To prove Item 3, 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 $\mathcal{A}$ reports estimates with max error $\alpha(\hat{d}, \beta)$ with probability $\geq 1-\beta$, a union bound implies the minimum over all T repetitions can only be wrong by $\alpha(\hat{d}, \beta/T)$ with probability $\geq 1-\beta$. Thus, it suffices to bound the probability that E_j does not occur for some j.

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

□

B.3 Communication-efficient Histograms & Range queries

In Table 5, we compare the above protocol with the protocols by Ghazi et al [36]. The communication complexities of those protocols have only a logarithmic dependence on n, d. They combine compression techniques that found success in the local model with the privacy blanket notion.

The table presents two other histogram protocols. The first uses the parallel-counts template that we used in Section 3.2, but now with the binary sum protocol presented by Ghazi, Kumar, Manurangsi, and Pagh [37]. The expected message complexity of this protocol vanishes with n, so that a large userbase counteracts a large dimension d. The second also has a vanishing message complexity, but with a faster rate. Each user in this protocol by Cheu and Zhilyaev [28] randomizes the one-hot encoding of their data, as well as a small number of $(0,\ldots,0)$ strings. These fake users contribute just enough cover noise to protect real users.

Table 5: Shuffle protocols for histograms. All take $\delta > 0$. We assume $\delta < 1/\log d$ for results from [36]. T is a natural number. The notation $\tilde{O}(\ldots)$ suppresses nested logarithms.

Source	Error	Messages per User	Bits per Message
Thm. 30	$O\left(\frac{T^2}{\varepsilon^2}\log\frac{T}{\delta}\right)$	$O\left(\frac{T^3 d^{1/T}}{\varepsilon^2}\log\frac{T}{\delta}\right)$	$O\left(\log Tn + \frac{1}{T}\log d\right)$
[36]	$\tilde{O}\left(\frac{1}{\varepsilon}\sqrt{\log^3 d \log\frac{1}{\delta}}\right)$	$\tilde{O}\left(\frac{1}{\varepsilon^2}\log^3 d \log\frac{1}{\delta}\right)$	$O(\log n + \log\log d)$
	$O\left(\log d + \frac{1}{\varepsilon}\sqrt{\log d \log\frac{1}{\varepsilon\delta}}\right)$	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon\delta}\right)$	$O(\log n \log d)$
[37]	$O(\frac{1}{\varepsilon}\log d)$	$1 + O(\frac{d}{\varepsilon^2 n}\log^2\frac{1}{\delta})$	$O(\log d)$
[28]	$O(\log d + \frac{1}{\varepsilon}\sqrt{\log d \log\frac{1}{\delta}})$	$1 + O(\frac{\log d}{n} + \frac{1}{\varepsilon^2 n}\log\frac{1}{\delta})$	d

In [36], Ghazi et al. 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 6 for a summary of the results.

Table 6: 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	$O(\frac{1}{\varepsilon}\log^{2d+3/2}(k^d)\log\frac{1}{\delta})$	$O(\frac{1}{\varepsilon^2}\log^{3d+3}(k^d)\log\frac{1}{\delta})$	$O(\log n + \log(d\log k))$
Hadamard	$O(\frac{1}{\varepsilon}\log^{2d+1/2}(k^d)\log\frac{1}{\varepsilon\delta})$	$O(\frac{1}{\varepsilon^2}\log^{2d}(k^d)\log\frac{1}{\varepsilon\delta})$	$O(\log(n)\cdot d\log k)$

B.4 A Lower Bound for Binary Sums

A result by Ghazi et al. [35] states that every communication-bounded shuffle protocol must imply some local protocol with a nontrivial privacy guarantee:

Lemma 32 (Ghazi et al. [35])**.** *Suppose $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ satisfies $(O(1), 0)$-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 33 (Ghazi et al. [35])**.** *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.5 A Lower Bound for Vector Sums

In the context of the secure aggregation model, Chan, Choquette-Choo, Kairouz, and Suresh [21] prove the following lemma regarding finite-precision representations of vectors:

Lemma 34. *Fix any algorithm M that takes as input a d-dimensional unit vector (in Euclidean space) and outputs b bits. If there exists an algorithm A where $\mathbb{E}\left[\|(A \circ M)(x) - x\|_2^2\right] \leq \alpha$, then $b \geq \frac{d}{2}\log_2(1/\alpha)$. If we also have that $\mathbb{E}\left[(A \circ M)(x) - x\right] = 0$, then $b = \Omega(d/\alpha)$.*

We can use the above to obtain a lower bound on the communication complexity of any secure intermediary protocol for vector sums

Corollary 35. *Let $P = (R, I, A)$ be any secure intermediary protocol (e.g. I is a shuffler or aggregator). If P privately estimates vector means with near-optimal ℓ_2 error—formally, $\mathbb{E}\left[\|P(\vec{x}) - \frac{1}{n}\sum x_i\|_2^2\right] = \tilde{O}(d/n^2\varepsilon^2)$—then R must be supported on a set of size at least 2^b for $b = \tilde{\Omega}(\max(d\log(n^2\varepsilon^2/d), 1))$. If the estimate is unbiased, then $b = \tilde{\Omega}(\min(d, n^2\varepsilon^2))$.*

The shuffle protocol by Cheu, Joseph, Mao, and Peng [23] is unbiased and has near-optimal error. It executes d scalar mean protocols in parallel, each one consuming $O(\sqrt{n} + \frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ messages of $O(\log d)$ bits. In the regime where $d < n^2\varepsilon^2$, the communication complexity is suboptimal.

To improve that bound, the authors suggest replacing the scalar mean subroutine with that of Balle Bell Gascón and Nissim. This alternative protocol is biased and, once we re-scale parameters and label messages for use in the vector mean protocol, consumes $O\left(\left(\log\frac{d}{\delta} + \log(n + \frac{1}{\varepsilon}\log d)\right)/\log n\right)$ messages of $\log(d(n + \frac{1}{\varepsilon}\log d))$ bits. So this modification is within polylogarithmic factors of the general lower bound

C 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 36. *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 $\{0,\dots,n+2\}$ 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 $G = \{0,\dots,n^2+2n\}$. 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 $G := \{0,\dots,n^2+2n\}$ 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: $\mathbb{P}[|(\mathcal{S} \circ \mathcal{R}^n)(0^n)| = 0] = \mathbb{P}[|\mathcal{R}(0)| = 0]^n > 0$. For the inductive step, we are given that $\mathbb{P}[|(\mathcal{S} \circ \mathcal{R}^n)(0^n)| = g] > 0$ for some $g \in G - \{n^2+2n\}$ and we show that $\mathbb{P}[|(\mathcal{S} \circ \mathcal{R}^n)(0^n)| = g+1] > 0$. There must be a vector $\vec{g} \in \{0,\dots,n+2\}^n$ 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 $\mathbb{P}[|(\mathcal{S} \circ \mathcal{R}^n)(0^n)| = g+1] \geq \prod_{j=1}^n \mathbb{P}\left[|\mathcal{R}(0)| = g_j'\right] > 0$.

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 $\mathrm{supp}(|\mathcal{R}(1)|) \subset \mathrm{supp}(|\mathcal{R}(0)|)$ implies

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

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 37. *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. □