Separating Local & Shuffled Differential Privacy via Histograms

Victor Balcer
School of Engineering & Applied Sciences, Harvard University, Cambridge, MA, United States

Albert Cheu
Khoury College of Computer Sciences, Northeastern University, Boston, MA, United States

Abstract

Recent work in differential privacy has highlighted the *shuffled model* as a promising avenue to compute accurate statistics while keeping raw data in users' hands. We present a protocol in this model that estimates histograms with error *independent of the domain size*. This implies an arbitrarily large gap in sample complexity between the shuffled and local models. On the other hand, we show that the models are equivalent when we impose the constraints of pure differential privacy and single-message randomizers.

2012 ACM Subject Classification Security and privacy → Privacy-preserving protocols

Keywords and phrases Differential Privacy, Distributed Protocols, Histograms

Digital Object Identifier 10.4230/LIPIcs.ITC.2020.1

Related Version A full version of the paper is available at `https://arxiv.org/abs/1911.06879`.

Funding *Victor Balcer*: Supported by NSF grant CNS-1565387.
Albert Cheu: Supported by NSF grants CCF-1718088, CCF-1750640, and CNS-1816028.

Acknowledgements We are grateful to Daniel Alabi and Maxim Zhilyaev for discussions that shaped the early stages of this work. We are also indebted to Matthew Joseph and Jieming Mao for directing us to the pointer-chasing and multi-party pointer-jumping problems. Finally, we thank Salil Vadhan for editorial comments and providing a simpler construction for Claim 19.

1 Introduction

The *local model* of private computation has minimal trust assumptions: each user executes a randomized algorithm on their data and sends the output *message* to an analyzer [17, 19, 11]. While this model has appeal to the users – their data is never shared in the clear – the noise in every message poses a roadblock to accurate statistics. For example, a locally private d-bin histogram has, on some bin, error scaling with $\sqrt{\log d}$. But when users trust the analyzer with their raw data (the *central model*), there is an algorithm that achieves error independent of d on every bin.

Because the local and central models lie at the extremes of trust, recent work has focused on the intermediate *shuffled model* [6, 8]. In this model, users execute ranomization like in the local model but now a trusted *shuffler* applies a uniformly random permutation to all user messages before the analyzer can view them. The anonymity provided by the shuffler allows users to introduce less noise than in the local model while achieving the same level of privacy. This prompts the following questions:

In terms of accuracy, how well separated is the shuffled model from the local model?

How close is the shuffled model to the central model?

1.1 Our Results

In Section 3, we provide a new protocol for histograms in the shuffled model. To quantify accuracy, we bound the *simultaneous error*: the maximum difference over all bins between each bin's estimated frequency and its true frequency in the input dataset.

▶ **Theorem 1** (Informal). *For any $\varepsilon < 1$ and $\delta = o(1/n)$, there exists a shuffled protocol that satisfies (ε, δ)-differential privacy and reports a histogram with simultaneous error $O(\log(1/\delta)/(\varepsilon^2 n))$ with constant probability.*

For comparison, [8] give a protocol with error $O(\sqrt{\log d \cdot \log 1/\delta}/(\varepsilon n))$. Our protocol has smaller error when $\log(1/\delta) = o(\log d)$. In the natural regime where $\delta = \Theta(poly(1/n))$, that condition is satisfied when $\log n = o(\log d)$. An example for this setting would be a dataset holding the browser home page of each user. The data universe could consist of all strings up to a certain length which would far exceed the number of users.

In Section 3.3, we show that the histogram protocol has strong implications for the *distributional* setting. Here, the rows of the dataset are independently drawn from a probability distribution. We focus on the *sample complexity*, which is the number of samples needed to identify or estimate some feature of the distribution. We prove that the separation in sample complexity between the local and shuffled models can be made arbitrarily large:

▶ **Theorem 2** (Informal). *There is a distributional problem where the sample complexity in the local model scales with a parameter of the problem, but the sample complexity in the shuffled model is independent of that parameter.*

We also show that there is a distributional problem which requires polynomially more samples in the *sequentially interactive* local model than in the shuffled model. This is done by reusing the techniques to prove Theorem 2.

A natural conjecture is that there are progressively weaker versions of Theorem 2 for progressively constrained versions of the model. In Section 4, we prove that the shuffled model collapses to the local model when constraints are too strong:

▶ **Theorem 3** (Informal). *For every single-message shuffled protocol that satisfies pure differential privacy, there is a local protocol with exactly the same privacy and sample complexity guarantees.*

1.2 Related Work

Table 1 presents our histogram result alongside existing results for the problem – all previous bounds on simultaneous error in the shuffled model depend on d. Although we focus on simultaneous error, error metrics focusing on per-bin error are also used in the literature such as mean squared error (MSE) and high probability confidence intervals on each bin. When considering these alternative metrics or when d is not large, other histogram protocols may outperform ours (see e.g. [18]).

Quantitative separations between the local and shuffled models exist in the literature [8, 1, 13, 12]. As a concrete example, [8] implies that the sample complexity of Bernoulli mean estimation in the shuffled model is $O(1/\alpha^2 + \log(1/\delta)/(\alpha\varepsilon))$. In contrast, [5] gives a lower bound of $\Omega(1/\alpha^2\varepsilon^2)$ in the local model.

Prior work have shown limits of the shuffled model, albeit under communication constraints. The first set of results follow from a lemma in [8]: a single-message shuffled protocol implies a local protocol with a weaker differential privacy guarantee. Specifically, if the shuffled protocol obeys (ε, δ)-differential privacy, then the local protocol obeys $(\varepsilon + \ln n, \delta)$-differential privacy. Lower bounds for the local model can then be invoked, as done in [8, 13].

Table 1 Comparison of results for the histogram problem. To simplify the presentation, we assume constant success probability, $\varepsilon < 1$, $\delta < 1/\log d$ for results from [13], and $e^{-O(n\varepsilon^2)} \leq \delta < 1/n$ for our result.

Model	Simultaneous Error	No. Messages per User	Source
Local	$\Theta\left(\frac{1}{\varepsilon\sqrt{n}} \cdot \sqrt{\log d}\right)$	1	[3]
Shuffled	$O\left(\frac{1}{\varepsilon n} \cdot \sqrt{\log d \cdot \log \frac{1}{\delta}}\right)$	$O(d)$	[8]
	$O\left(\frac{1}{\varepsilon n}\sqrt{\log^3 d \cdot \log(\frac{1}{\delta}\log d)}\right)$	$O\left(\frac{1}{\varepsilon^2}\log^3 d \cdot \log(\frac{1}{\delta}\log d)\right)$ w.h.p.	[13]
	$O\left(\frac{\log d}{n} + \frac{1}{\varepsilon n} \cdot \sqrt{\log d \cdot \log \frac{1}{\varepsilon\delta}}\right)$	$O\left(\frac{1}{\varepsilon^2}\log \frac{1}{\varepsilon\delta}\right)$	[13]
	$O\left(\frac{1}{\varepsilon^2 n}\log \frac{1}{\delta}\right)$	$O(d)$	[Theorem 12]
Central	$\Theta\left(\frac{1}{\varepsilon n}\min\left(\log d, \log \frac{1}{\delta}\right)\right)$	N/A	[9, 7, 4, 14]

Another class of lower bound comes from a lemma in [12]: when a shuffled protocol obeys ε-differential privacy and bounded communication complexity, the set of messages output by each user is insensitive to their personal data. Specifically, changing their data causes the set's distribution to change by $\leq 1 - 2^{-O_\varepsilon(m^2\ell)}$ in statistical distance, where m denotes number of messages and ℓ the length of each message. This is strong enough to obtain a lower bound on binary sums.

The amplification-by-shuffling lemmas in [2, 10] show that uniformly permuting the messages generated by a local protocol improves privacy guarantees: an ε-private local protocol becomes an (ε', δ)-private shuffled protocol where $\varepsilon' \ll \varepsilon$ and $\delta > 0$. One might conjecture weaker versions of these lemmas where $\delta = 0$ but Theorem 3 eliminates that possibility.

2 Preliminaries

We define a *dataset* $\vec{x} \in \mathcal{X}^n$ to be 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* (denoted as $\vec{x} \sim \vec{x}\,'$) if they differ in exactly one row.

▶ **Definition 4** (Differential Privacy [9]). *An algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{Z}$ satisfies* (ε, δ)-differential privacy *if*

$$\forall \vec{x} \sim \vec{x}\,' \quad \forall T \subseteq \mathcal{Z} \qquad \Pr[\mathcal{M}(\vec{x}) \in T] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(\vec{x}\,') \in T] + \delta.$$

We say an (ε, δ)-differentially private algorithm satisfies pure differential privacy *when $\delta = 0$ and* approximate differential privacy *when $\delta > 0$. For pure differential privacy, we may omit the δ parameter from the notation.*

▶ **Definition 5** (Local Model [17]). *A protocol $\mathcal{P}$ in the* (non-interactive[1]) local model *consists of two randomized algorithms:*

- *A* randomizer $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ *that takes as input a single user's data and outputs a message.*

[1] The literature also includes interactive variants; see [15] for a definition of *sequential* and *full* interactivity.

- *An* analyzer $\mathcal{A} : \mathcal{Y}^* \to \mathcal{Z}$ *that takes as input all user messages and computes the output of the protocol.*

We denote the protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$*. We assume that the number of users* n *is public and available to both* $\mathcal{R}$ *and* $\mathcal{A}$*. Let* $\vec{x} \in \mathcal{X}^n$*. The evaluation of the protocol* $\mathcal{P}$ *on input* $\vec{x}$ *is The evaluation of the protocol* $\mathcal{P}$ *on input* $\vec{x}$ *is*

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

▶ **Definition 6** (Differential Privacy for Local Protocols)**.** *A local protocol* $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ *satisfies* (ε, δ)-differential privacy *for* n *users if its randomizer* $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ *is* (ε, δ)*-differentially private (for datasets of size one).*

▶ **Definition 7** (Shuffled Model [6, 8])**.** *A protocol* $\mathcal{P}$ *in the* shuffled model *consists of three randomized algorithms:*

- *A* randomizer $\mathcal{R} : \mathcal{X} \to \mathcal{Y}^*$ *that takes as input a single user's data and outputs a vector of messages whose length may be randomized. If, on all inputs, the probability of sending a single message is 1, then the protocol is said to be* single-message*. Otherwise, the protocol is said to be* multi-message*.*
- *A* shuffler $\mathcal{S} : \mathcal{Y}^* \to \mathcal{Y}^*$ *that concatenates all message vectors and then applies a uniformly random permutation to (the order of) the concatenated vector. For example, when there are three users each sending two messages, there are* 6! *permutations and all are equally likely to be the output of the shuffler.*
- *An* analyzer $\mathcal{A} : \mathcal{Y}^* \to \mathcal{Z}$ *that takes a permutation of messages to generate the output of the protocol.*

As in the local model, we denote the protocol $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ *and assume that the number of users* n *is accessible to both* $\mathcal{R}$ *and* $\mathcal{A}$*. The evaluation of the protocol* $\mathcal{P}$ *on input* $\vec{x}$ *is*

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

▶ **Definition 8** (Differential Privacy for Shuffled Protocols [8])**.** *A shuffled protocol* $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ *satisfies* (ε, δ)-differential privacy *for* n *users if the algorithm* $(\mathcal{S} \circ \mathcal{R}) : \mathcal{X}^n \to \mathcal{Y}^*$ *is* (ε, δ)*-differentially private.*

We note a difference in robustness between the local and shuffled models. A user in a local protocol only has to trust that their own execution of $\mathcal{R}$ is correct to ensure differential privacy. In contrast, a user in a shuffled protocol may not have the same degree of privacy when other users deviate from the protocol.

For any $d \in \mathbb{N}$, let $[d]$ denote the set $\{1, \ldots, d\}$. For any $j \in [d]$, we define the function $c_j : [d]^n \to \mathbb{R}$ as the normalized count of j in the input:

$$c_j(\vec{x}) = (1/n) \cdot |\{i \in [n] \,:\, x_i = j\}|.$$

We use *histogram* to refer to the vector of normalized counts $(c_1(\vec{x}), \ldots, c_d(\vec{x}))$. For measuring the accuracy of a histogram protocol $\mathcal{P} : [d]^n \to \mathbb{R}^d$, we use the following metrics:

▶ **Definition 9.** *A histogram protocol* $\mathcal{P} : [d]^n \to \mathbb{R}^d$ *has* (α, β)-per-query accuracy *if*

$$\forall \vec{x} \in [d]^n \quad \forall j \in [d] \quad \Pr[|\mathcal{P}(\vec{x})_j - c_j(\vec{x})| \leq \alpha] \geq 1 - \beta.$$

▶ **Definition 10.** *A histogram protocol* $\mathcal{P} : [d]^n \to \mathbb{R}^d$ *has* (α, β)-simultaneous accuracy *if*

$$\forall \vec{x} \in [d]^n \quad \Pr[\forall j \in [d] \quad |\mathcal{P}(\vec{x})_j - c_j(\vec{x})| \leq \alpha] \geq 1 - \beta.$$

3 The Power of Multiple Messages for Histograms

In this section, we present an (ε, δ)-differentially private histogram protocol in the shuffled model whose simultaneous error does not depend on the universe size. We start by presenting a private protocol for releasing a binary sum that always outputs 0 if the true count is 0 and otherwise outputs a noisy estimate. The histogram protocol uses this counting protocol to estimate the frequency of every domain element. Its simultaneous error is the maximum noise introduced to the nonzero counts. There are at most n such counts.

For comparison, a protocol in [8] adds independent noise to all counts without the zero-error guarantee. The simultaneous error is therefore the maximum noise over *all* d counts, which introduces a $\log d$ instead of a $\log n$ factor.

3.1 A Two-Message Protocol for Binary Sums

In the protocol $\mathcal{P}^{\text{zsum}}_{\varepsilon,\delta}$ (Figure 1), each user reports a vector whose length is the sum of their data and a Bernoulli random variable. The contents of each vector will be copies of 1. Because the shuffler only reports a uniformly random permutation, the observable information is equivalent to the sum of user data, plus noise. The noise is distributed as $\text{Bin}(n, p)$, where p is chosen so that there is sufficient variance to ensure (ε, δ)-differential privacy. We take advantage of the fact that the binomial distribution is bounded: if the sum of the data is zero, the noisy sum is *never* more than n. Hence, the analyzer will perform truncation when the noisy sum is small. We complete our proof by arguing that it is unlikely for large values to be truncated.

To streamline the presentation and analysis, we assume that $\sqrt{(100/n) \cdot \ln(2/\delta)} \leq \varepsilon \leq 1$ so that $p \in (1/2, 1)$. We can achieve (ε, δ) privacy for a broader parameter regime by setting p to a different function; we refer the interested reader to Theorem 4.11 in [8].

Randomizer $\mathcal{R}^{\text{zsum}}_{\varepsilon,\delta}(x \in \{0,1\})$ for $\varepsilon, \delta \in [0,1]$**:**
1. Let $p \leftarrow 1 - \frac{50}{\varepsilon^2 n} \ln(2/\delta)$.
2. Sample $z \sim \text{Ber}(p)$.
3. Output $(\underbrace{1, \ldots, 1}_{x+z \text{ copies}})$.

Analyzer $\mathcal{A}^{\text{zsum}}_{\varepsilon,\delta}(\vec{y} \in \{1\}^*)$ for $\varepsilon, \delta \in [0,1]$**:**
1. Let $p \leftarrow 1 - \frac{50}{\varepsilon^2 n} \ln(2/\delta)$.
2. Let $c^* = \frac{1}{n} \cdot |\vec{y}|$, where $|\vec{y}|$ is the length of $\vec{y}$.
3. Output $\begin{cases} c^* - p & \text{if } c^* > 1 \\ 0 & \text{otherwise} \end{cases}$.

Figure 1 The pseudocode for $\mathcal{P}^{\text{zsum}}_{\varepsilon,\delta}$, a private shuffled protocol for normalized binary sums.

▶ **Theorem 11.** *For any $\varepsilon, \delta \in [0,1]$ and any $n \in \mathbb{N}$ such that $n \geq (100/\varepsilon^2) \cdot \ln(2/\delta)$, the protocol $\mathcal{P}^{\text{zsum}}_{\varepsilon,\delta} = (\mathcal{R}^{\text{zsum}}_{\varepsilon,\delta}, \mathcal{A}^{\text{zsum}}_{\varepsilon,\delta})$ has the following properties:*

(i) *$\mathcal{P}^{\text{zsum}}_{\varepsilon,\delta}$ is (ε, δ)-differentially private in the shuffled model.*

(ii) *For every* $\beta \geq \delta^{25}$ *, the error is* $|\mathcal{P}^{\text{zsum}}_{\varepsilon,\delta}(\vec{x}) - \frac{1}{n}\sum x_i| \leq \alpha$ *with probability* $\geq 1-\beta$ *where*

$$\begin{aligned}\alpha &= \frac{50}{\varepsilon^2 n}\log\frac{2}{\delta} + \frac{1}{\varepsilon n}\cdot\sqrt{200\log\frac{2}{\delta}\cdot\log\frac{2}{\beta}} \\ &= O\left(\frac{1}{\varepsilon^2 n}\log\frac{1}{\delta}\right)\end{aligned}$$

(iii) $\mathcal{P}^{\text{zsum}}_{\varepsilon,\delta}(\underbrace{(0,\ldots,0)}_{n \text{ copies}}) = 0.$

(iv) *Each user sends at most two one-bit messages.*

Proof of Part (i). If we let z_i be the random bit generated by the i-th user, the total number of messages is $|\vec{y}| = \sum_{i=1}^n x_i + z_i$. Observe that learning $|\vec{y}|$ is sufficient to represent the output of shuffler since all messages have the same value. Thus, the privacy of this protocol is equivalent to the privacy of

$$\mathcal{M}(\vec{x}) = \sum_{i=1}^n x_i + \text{Bin}(n,p) \sim -\left(-\sum_{i=1}^n x_i + \text{Bin}(n,1-p)\right) + n.$$

By post-processing, it suffices to show the privacy of $\mathcal{M}_{\text{neg}}(\vec{x}) = -\sum_{i=1}^n x_i + \text{Bin}(n,1-p)$ where $1-p = \frac{50}{\varepsilon^2 n}\ln\frac{2}{\delta}$. Because privacy follows almost immediately from technical claims in [13], we defer the proof to Appendix A. ◀

Proof of Part (ii). Fix any $\vec{x} \in \{0,1\}^n$. For shorthand, we define $\alpha' = 2\cdot\sqrt{\frac{p(1-p)}{n}\cdot\ln(2/\beta)}$ so that $\alpha = (1-p) + \alpha'$. A Chernoff bound implies that for $\beta \geq 2e^{-np(1-p)}$, the following event occurs with probability $\geq 1-\beta$:

$$\left|\frac{1}{n}\cdot\sum_{i=1}^n z_i - p\right| \leq \alpha' \tag{1}$$

The inequality $\beta \geq 2e^{-np(1-p)}$ follows from our bounds on ε, β, and n.

The remainder of the proof will condition on (1). If $c^* > 1$, then the analyzer outputs $c^* - p$. We show that the error of $c^* - p$ is at most α':

$$\begin{aligned}\left|(c^*-p) - \frac{1}{n}\cdot\sum_{i=1}^n x_i\right| &= \left|\frac{1}{n}\cdot\sum_{i=1}^n (x_i+z_i) - p - \frac{1}{n}\cdot\sum_{i=1}^n x_i\right| && \text{(By construction)} \\ &= \left|\frac{1}{n}\cdot\sum_{i=1}^n z_i - p\right| \\ &\leq \alpha' && \text{(By (1))}\end{aligned}$$

If $c^* \leq 1$, then the analyzer will output 0. In this case, the error is exactly $\frac{1}{n}\sum x_i$. We argue that $c^* \leq 1$ implies $\frac{1}{n}\sum x_i \leq \alpha$.

$$\begin{aligned}1 &\geq c^* \\ &= \frac{1}{n}\cdot\sum_{i=1}^n (x_i+z_i) && \text{(By construction)} \\ &\geq \frac{1}{n}\cdot\sum_{i=1}^n x_i + p - \alpha' && \text{(By (1))}\end{aligned}$$

Rearranging terms yields

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

which concludes the proof. ◀

Proof of Part (iii). If $\vec{x} = (0, \ldots, 0)$, then $|\vec{y}|$ is drawn from $0 + \mathrm{Bin}(n, p)$, which implies $c^* \leq 1$ with probability 1. Hence, $\mathcal{P}^{\mathrm{zsum}}_{\varepsilon,\delta}(\vec{x}) = 0$. ◀

3.2 A Multi-Message Protocol for Histograms

In the protocol $\mathcal{P}^{\mathrm{hist}}_{\varepsilon,\delta}$ (Figure 2), users encode their data $x_i \in [d]$ as a one-hot vector $\vec{b} \in \{0,1\}^d$. Then protocol $\mathcal{P}^{\mathrm{zsum}}_{\varepsilon,\delta}$ is executed on each coordinate j of $\vec{b}$. The executions are done in one round of shuffling. To remove ambiguity between executions, each message in execution j has value j.

Randomizer $\mathcal{R}^{\mathrm{hist}}_{\varepsilon,\delta}(x \in [d])$ for $\varepsilon, \delta \in [0,1]$**:**
1. For each $j \in [d]$, let $b_j \leftarrow \mathbb{1}[x = j]$ and compute scalar product $\vec{m}_j \leftarrow j \cdot \mathcal{R}^{\mathrm{zsum}}_{\varepsilon,\delta}(b_j)$.
2. Output the concatenation of all $\vec{m}_j$.

Analyzer $\mathcal{A}^{\mathrm{hist}}_{\varepsilon,\delta}(\vec{y} \in [d]^*)$ for $\varepsilon, \delta \in [0,1]$**:**
1. For each $j \in [d]$, let $\vec{y}_{(j)} \leftarrow$ all messages of value j, then compute $\tilde{c}_j \leftarrow \mathcal{A}^{\mathrm{zsum}}_{\varepsilon,\delta}(\vec{y}_{(j)})$.
2. Output $(\tilde{c}_1, \ldots, \tilde{c}_d)$.

Figure 2 The pseudocode for $\mathcal{P}^{\mathrm{hist}}_{\varepsilon,\delta}$, a private shuffled protocol for histograms.

▶ **Theorem 12.** *For any $\varepsilon, \delta \in [0,1]$ and any $n \in \mathbb{N}$ such that $n \geq (100/\varepsilon^2) \cdot \ln(2/\delta)$, the protocol $\mathcal{P}^{\mathrm{hist}}_{\varepsilon,\delta} = (\mathcal{R}^{\mathrm{hist}}_{\varepsilon,\delta}, \mathcal{A}^{\mathrm{hist}}_{\varepsilon,\delta})$ has the following properties:*

(i) *$\mathcal{P}^{\mathrm{hist}}_{\varepsilon,\delta}$ is $(2\varepsilon, 2\delta)$-differentially private in the shuffled model.*

(ii) *For every $\beta \geq \delta^{25}$, $\mathcal{P}^{\mathrm{hist}}_{\varepsilon,\delta}$ has (α, β)-per-query accuracy for*

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

(iii) *For every $\beta \geq n \cdot \delta^{25}$, $\mathcal{P}^{\mathrm{hist}}_{\varepsilon,\delta}$ has (α, β)-simultaneous accuracy for*

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

(iv) *Each user sends at most $1 + d$ messages each of length $O(\log d)$.*

The accuracy guaranteed by this protocol is close to what is possible in the central model: there is a stability-based algorithm with simultaneous error $O((1/(\varepsilon n)) \cdot \ln(1/\delta))$ [7]. However, in $\mathcal{P}^{\mathrm{hist}}_{\varepsilon,\delta}$, each user communicates $O(d)$ messages of $O(\log d)$ bits. It remains an open question as to whether or not this can be improved while maintaining similar accuracy.

Because the simultaneous error of a *single-message* histogram protocol is at least $\Omega((1/(\varepsilon n)) \cdot \mathrm{poly}(\log d))$ [8], this protocol is also proof that the single-message model is

a strict subclass of the multi-message model. This separation was previously shown by [2, 1] for the summation problem.[2]

Proof of Part (i). Fix any neighboring pair of datasets $\vec{x} \sim \vec{x}\,'$. Let $\vec{y} \leftarrow (\mathcal{S} \circ \mathcal{R}^{\text{hist}}_{\varepsilon,\delta})(\vec{x})$ and $\vec{y}\,' \leftarrow (\mathcal{S} \circ \mathcal{R}^{\text{hist}}_{\varepsilon,\delta})(\vec{x}\,')$. For any $j \neq j'$, the count of j in output of the shuffler is independent of the count of j' in the output because each execution of $\mathcal{R}^{\text{zsum}}_{\varepsilon,\delta}$ is independent. As in Step (1) of $\mathcal{A}^{\text{hist}}_{\varepsilon,\delta}$, for $j \in [d]$, let $\vec{y}_{(j)}$ ($\vec{y}\,'_{(j)}$ resp.) be the vector of all messages in $\vec{y}$ ($\vec{y}\,'$ resp.) that have value j.

For any $j \in [d]$ where $c_j(\vec{x}) = c_j(\vec{x}\,')$, $\vec{y}_{(j)}$ is identically distributed to $\vec{y}\,'_{(j)}$. For each of the two $j \in [d]$ where $c_j(\vec{x}) \neq c_j(\vec{x}\,')$, we will show that the distribution of $\vec{y}_{(j)}$ is close to that of $\vec{y}\,'_{(j)}$. Let $\vec{r}, \vec{r}\,' \in \{0,1\}^n$ where $r_i = \mathbb{1}[x_i = j]$ and $r'_i = \mathbb{1}[x'_i = j]$. Now,

$$\vec{y}_{(j)} \sim j \cdot (\mathcal{S} \circ \mathcal{R}^{\text{zsum}}_{\varepsilon,\delta})(\vec{r}) \quad \text{and} \quad \vec{y}\,'_{(j)} \sim j \cdot (\mathcal{S} \circ \mathcal{R}^{\text{zsum}}_{\varepsilon,\delta})(\vec{r}\,').$$

So by Theorem 11 Part (i), for any $T \subseteq \{j\}^*$,

$$\Pr[\vec{y}_{(j)} \in T] \leq e^{\varepsilon} \cdot \Pr[\vec{y}\,'_{(j)} \in T] + \delta.$$

$(2\varepsilon, 2\delta)$-differential privacy follows by composition. ◀

Proof of Part (ii)-(iii). Notice that the j-th element in the output $\tilde{c}_j$ is identically distributed with an execution of the counting protocol on the bits $b_{i,j}$ indicating if $x_i = j$. Formally, $\tilde{c}_j \sim \mathcal{P}^{\text{zsum}}_{\varepsilon,\delta}(\{b_{i,j}\}_{i\in[n]})$ for all $j \in [d]$. Per-query accuracy immediately follows from Theorem 11 Part (ii).

To bound simultaneous error, we leverage the property that when $c_j(\vec{x}) = 0$, the counting protocol will report a nonzero value with probability 0. Let $Q = \{j \in [d] : c_j(\vec{x}) > 0\}$ and let α be the error bound defined in Theorem 11 Part (ii) for the failure probability β/n.

$$\begin{aligned}
&\Pr\left(\exists j \in [d] \text{ s.t. } |\tilde{c}_j - c_j(\vec{x})| > \alpha\right) \\
\leq &\Pr\left(\exists j \in Q \text{ s.t. } |\tilde{c}_j - c_j(\vec{x})| > \alpha\right) + \Pr\left(\exists j \notin Q \text{ s.t. } |\tilde{c}_j - c_j(\vec{x})| > \alpha\right) \\
= &\Pr\left(\exists j \in Q \text{ s.t. } |\tilde{c}_j - c_j(\vec{x})| > \alpha\right) && \text{(Theorem 11 Part (iii))} \\
\leq &\sum_{j \in Q} \Pr\left(|\tilde{c}_j - c_j(\vec{x})| > \alpha\right) \\
\leq &\sum_{j \in Q} \beta/n && \text{(Theorem 11 Part (ii))} \\
\leq &\beta && (|Q| \leq n)
\end{aligned}$$

This concludes the proof. ◀

3.3 Applications

In this section, we use our histogram protocol to solve two distributional problems; one of these results implies a very strong separation in sample complexity between the non-interactive local model and the shuffled model. Both distributional problems reduce to what we call *support identification*:

[2] In particular, a private unbiased estimator for $\sum_i x_i$ with real-valued $x_i \in [0,1]$ in the single-message shuffled model must have error $\Omega(n^{1/6})$ [2] while there exists a multi-message shuffled model protocol for estimating summation with error $O(1/\varepsilon)$ [1].

▶ **Definition 13** (Support Identification Problem). *The* support identification *problem has positive integer parameters $h \leq d$. Let D be a set of size d and let $\mathbf{U}_H$ be the uniform distribution over any $H \subseteq D$. The set of problem instances is $\{\mathbf{U}_H : H \subseteq D \text{ and } |H| = h\}$. A protocol solves the* (h, d)-support identification problem with sample complexity n *if, given n users with data independently sampled from any problem instance $\mathbf{U}_H$, it identifies H with probability at least 99/100.*

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

▷ Claim 14. Fix any $\varepsilon \in (0, 1]$ and $\delta < (1/200h)^{1/25}$. Under (ε, δ)-differential privacy, the sample complexity of the (h, d)-support identification problem is $O(h \log h \cdot (1/\varepsilon^2) \cdot \log(1/\delta))$ in the shuffled model.

Proof. For the purposes of this proof, we assume there is some bijection f between D and $[d]$ so that any reference to $j \in [d]$ corresponds directly to some $f(j) \in D$ and vice versa. Consider the following protocol: execute $\mathcal{P}^{\text{hist}}_{\varepsilon,\delta}$ on n samples from $\mathbf{U}_H$ and then choose the items whose estimated frequencies are at least $(t + 1)/n$ (the magnitude of t will be determined later). We will prove that the items returned by the protocol are precisely those of H, with probability at least 99/100.

Let E_{samp} be the event that every element in support H has frequency at least $(2t+1)/n$ in the sample. Let E_{priv} be the event that the histogram protocol estimates the frequency of every element in D with error at most t/n. If both events occur, every element in H has estimated frequency at least $(t + 1)/n$ and every element outside H has estimated frequency at most t/n. Hence, it suffices to show that E_{samp} and E_{priv} each occur with probability $\geq 199/200$.

We lower bound the probability of E_{samp} via a coupon collector's argument. That is, if we have $n = O(kh \log h)$ samples from $\mathbf{U}_H$ then each element of H appears at least k times with probability at least 199/200. Hence we set $k = (2t + 1)$.

To lower bound the probability of E_{priv}, we simply invoke Theorem 12: given that $\varepsilon \in (0, 1]$ and $\delta > (1/200h)^{1/25}$, the frequency of every item in D is estimated up to error t/n for some $t = O((1/\varepsilon^2) \cdot \log(1/\delta))$ with probability $\geq 199/200$.[3] ◁

In the above analysis, if we had used a protocol with simultaneous error that depends on the domain size d, then t would in turn depend on d. For example, using the histogram protocol in [8] would give $t = \Omega((1/\varepsilon) \cdot \sqrt{\log d \cdot \log(1/\delta)})$. This results in a protocol whose sample complexity grows with d in addition to h.

So having shown how to solve the support identification problem with few samples, we now describe two different problems and explain how to reduce these to support identification. This will imply low sample complexity in the shuffled model.

▶ **Definition 15** (Pointer-Chasing Problem [16]). *The* pointer chasing problem *is denoted* $\text{PC}(k, \ell)$ *where k, ℓ are positive integer parameters. A problem instance is $\mathbf{U}_{\{(1,a),(2,b)\}}$ where a, b are permutations of $[\ell]$. A protocol* solves $\text{PC}(k, \ell)$ with sample complexity n *if, given n independent samples from any $\mathbf{U}_{\{(1,a),(2,b)\}}$, it outputs the k-th integer in the sequence $a_1, b_{a_1}, a_{b_{a_1}} \ldots$ with probability at least* 99/100.

To solve $\text{PC}(k, \ell)$, note that it suffices to identify $\{(1, a), (2, b)\}$ and directly perform the pointer chasing. Because the support has size $h = 2$, $\mathcal{P}^{\text{hist}}_{\varepsilon,\delta}$ can be used to solve the problem with just $O((1/\varepsilon^2) \cdot \log(1/\delta))$ samples, independent of k and ℓ. But in the case

[3] The bound on δ in Theorem 12 is a function of n. This is derived from a pessimistic bound on the number of unique values in the input. But in this reduction, we know that data takes one of h values.

where $k = 2$, [16] gives a lower bound of $\Omega(\ell/e^{\varepsilon})$ for non-interactive local protocols. So there is an arbitrarily large separation between the non-interactive shuffled and non-interactive local models (Theorem 2).

▶ **Definition 16** (Multi-Party Pointer Jumping Problem [15]). *The* multi-party pointer jumping problem *is denoted* $\mathrm{MPJ}(s,h)$ *where* s, h *are positive integer parameters. A problem instance is* $\mathbf{U}_{\{Z_1,\ldots,Z_h\}}$ *where each* Z_i *is a labeling of the nodes at level* i *in a complete* s*-ary tree. Each label* $Z_{i,j}$ *is an integer in* $\{0,\ldots,s-1\}$*. The labeling implies a root-leaf path: if the* i*-th node in the path has label* $Z_{i,j}$*, then the* $(i+1)$*-st node in the path is the* $(Z_{i,j})$*-th child of the* i*-th node. A protocol* solves $\mathrm{MPJ}(s,h)$ with sample complexity n *if, given* n *samples from any* $\mathbf{U}_{\{Z_1,\ldots,Z_h\}}$*, it identifies the root-leaf path with probability at least* $99/100$*.*

As with pointer-chasing, we can solve $\mathrm{MPJ}(s,h)$ when the support is identified. This takes $O(h\log h\cdot(1/\varepsilon^2)\cdot\log(1/\delta))$ samples in the shuffled model. But [15] gives a lower bound of $\Omega(h^3/(\varepsilon^2\log h))$ in the local model when $s = h^4$, even allowing for sequential interactivity. However, we do not claim a polynomial separation between the shuffled model and sequentially interactive local model. This would require a proof that every sequentially interactive local protocol has a counterpart in the shuffled model.

Note that the reductions we employ can also be applied in the central model. That is, instead of executing $\mathcal{P}^{\mathrm{hist}}_{\varepsilon,\delta}$ in the reduction (Claim 14), execute the central model algorithm, from [7], with simultaneous error $O((1/(\varepsilon n))\cdot\log(1/\delta))$. This improves the bounds by $1/\varepsilon$.

Table 2 The sample complexity of private pointer-chasing (PC) and multi-party pointer jumping (MPJ). Shuffled and central results follow from a reduction to histograms.

Model	$\mathrm{PC}(k,\ell)$	$\mathrm{MPJ}(s,h)$
Local	$\Omega(\ell/e^{\varepsilon})$ [16] for $k=2$	$\Omega(h^3/(\varepsilon^2\log h))$ [15] for $s=h^4$, seq. interactive
Shuffled	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$	$O\left(h\log h\cdot\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$ for $\delta<(1/200h)^{1/25}$
Central	$O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\right)$	$O\left(h\log h\cdot\frac{1}{\varepsilon}\log\frac{1}{\delta}\right)$

4 Pure Differential Privacy in the Shuffled Model

In this section, we prove that any single-message shuffled protocol that satisfies ε-differential privacy can be simulated by a local protocol under the same privacy constraint.

▶ **Theorem 17** (Formalization of Thm. 3). *For any single-message shuffled protocol* $\mathcal{P} = (\mathcal{R},\mathcal{A})$ *that satisfies* ε*-differential privacy, there exists a local protocol* $\mathcal{P}_L = (\mathcal{R}_L,\mathcal{A}_L)$ *that satisfies* ε*-differential privacy and* $\mathcal{P}_L(\vec{x})$ *is identically distributed to* $\mathcal{P}(\vec{x})$ *for every input* $\vec{x}\in\mathcal{X}^n$*.*

We start with the following claim, which strengthens a theorem in [8] for the special case of pure differential privacy in the shuffled model:

▷ Claim 18. Let $\mathcal{P} = (\mathcal{R},\mathcal{A})$ be any single-message shuffled protocol that satisfies ε-differential privacy. Then $\mathcal{R}$ is an ε-differentially private algorithm.

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

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

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

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

which contradicts the fact that $\mathcal{S} \circ \mathcal{R}$ is ε-differentially private. ◁

Now we are ready to prove Theorem 17.

Proof of Theorem 17. Consider the aggregator $\mathcal{A}_L$ that applies a uniformly random permutation to its input and then executes $\mathcal{A}$. Then $\mathcal{P}_L = (\mathcal{R}, \mathcal{A}_L)$ is a local protocol that simulates $\mathcal{P}$, in the sense that $\mathcal{P}_L(\vec{x})$ is identically distributed to $\mathcal{P}(\vec{x})$ for every $\vec{x} \in \mathcal{X}^n$. And by Claim 18, the randomizer is ε-differentially private. ◀

4.1 Roadblocks to Generalizing Theorem 17

One might conjecture Claim 18 also holds for multi-message protocols and thus immediately generalize Theorem 17. However, this is not the case:

▷ Claim 19. There exists a multi-message shuffled protocol that is ε-differentially private for all $\varepsilon \geq 0$ but its randomizer is not ε-differentially private for *any* finite ε.

Proof. Consider the randomizer $\mathcal{R}^{\infty}$ that on input $x \in \{0, 1\}$ outputs two messages x and $1 - x$. The output of the shuffler $S \circ \mathcal{R}^{\infty}$ is 0-differentially private since for all inputs the output is a random permutation of exactly n 0s and n 1s. However, $\mathcal{R}^{\infty}$ is not ε-differentially private for any finite ε as the first message of $\mathcal{R}^{\infty}(x)$ is that user's bit x. ◁

We note that it is without loss of accuracy or privacy to suppose that a randomizer shuffles its messages prior to sending them to the shuffler. We call these *pre-shuffle* randomizers. Observe that the pre-shuffle version of $\mathcal{R}^{\infty}$ (i.e. $\mathcal{S} \circ \mathcal{R}^{\infty}$ for 1 user) satisfies 0-differential privacy. So one might conjecture Claim 18 holds for pre-shuffle randomizers and thus generalize Theorem 17. But this too is not the case:

▷ Claim 20. There exists a multi-message shuffled protocol that is ε-differentially private for some finite ε but its pre-shuffle randomizer is not ε-differentially private for *any* finite ε.

Proof. Consider any randomizer $\mathcal{R}^{\text{gap}}$ that takes binary input and outputs four binary messages with the following constraint: the messages can take any value when the input is 0 but on input 1, there cannot be exactly two 1s. Formally, the supports are $\text{supp}(\mathcal{R}^{\text{gap}}(0)) = \{0, 1\}^4$ and $\text{supp}(\mathcal{R}^{\text{gap}}(1)) = \{\vec{y} \in \{0, 1\}^4 : \sum_i y_i \neq 2\}$.

The pre-shuffle randomizer $\mathcal{S} \circ \mathcal{R}^{\text{gap}}$ cannot satisfy pure differential privacy because $(0, 0, 1, 1) \in \text{supp}(\mathcal{R}^{\text{gap}}(0))$ but $(0, 0, 1, 1) \notin \text{supp}(\mathcal{R}^{\text{gap}}(1))$. On the other hand, for all $n \geq 2$ and $\vec{x} \in \{0, 1\}^n$,

$$\text{supp}(\mathcal{S} \circ \mathcal{R}^{\text{gap}}(\vec{x})) = \{0, 1\}^{4n}$$

This follows from the fact that every number in $\{0, \ldots, 4n\}$ – the number of 1s sent to the shuffler – can be expressed as the sum of n numbers from $\{0, 1, 3, 4\}$. Thus, there is some finite ε for which the protocol with randomizer $\mathcal{R}^{\text{gap}}$ is ε-differentially private. ◁

References

1 Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. Differentially private summation with multi-message shuffling. *arXiv preprint arXiv:1906.09116*, 2019.

2 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. doi:10.1007/978-3-030-26951-7_22.

3 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. doi:10.1145/2746539.2746632.

4 Amos Beimel, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 437–454. Springer, 2010. doi:10.1007/978-3-642-11799-2_26.

5 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. doi:10.1007/978-3-540-85174-5_25.

6 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. doi:10.1145/3132747.3132769.

7 Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 369–380. ACM, 2016. doi:10.1145/2840728.2840747.

8 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. doi:10.1007/978-3-030-17653-2_13.

9 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. doi:10.1007/11681878_14.

10 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. doi:10.1137/1.9781611975482.151.

11 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. doi:10.1145/773153.773174.

12 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.

13 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.

14 Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 705–714. ACM, 2010. doi:10.1145/1806689.1806786.

15 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. doi:10.1109/FOCS.2019.00015.

16 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. doi:10.1137/1.9781611975994.31.

17 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. doi:10.1109/FOCS.2008.27.

18 Tianhao Wang, Min Xu, Bolin Ding, Jingren Zhou, Ninghui Li, and Somesh Jha. Practical and robust privacy amplification with multi-party differential privacy. *arXiv preprint arXiv:1908.11515*, 2019.

19 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 Privacy via Smooth Distributions

Ghazi, Golowich, Kumar, Pagh and Velingker [13] identify a class of distributions and argue that, if η is sampled from such a distribution, adding η to a 1-sensitive sum ensures differential privacy of that sum.

▶ **Definition 21** (Smooth Distributions, [13]). *A distribution* $\mathbf{D}$ *over* $\mathbb{Z}$ *is* (ε, δ, k)*-smooth if for all* $k' \in [-k, k]$*,*

$$\Pr_{Y \sim \mathbf{D}}\left[\frac{\Pr_{Y' \sim \mathbf{D}}[Y' = Y]}{\Pr_{Y' \sim \mathbf{D}}[Y' = Y + k']} \geq e^{|k'|\varepsilon}\right] \leq \delta.$$

▶ **Lemma 22** (Smoothness for Privacy, [13]). *Let* $f : \mathbb{Z}^n \to \mathbb{Z}$ *be a function such that* $|f(\vec{x}) - f(\vec{x}')| \leq 1$ *for all* $\vec{x} \sim \vec{x}'$*. Let* $\mathbf{D}$ *be an* $(\varepsilon, \delta, 1)$*-smooth distribution. The algorithm that takes as input* $\vec{x} \in \mathbb{Z}^n$*, then samples* $\eta \sim \mathbf{D}$ *and reports* $f(\vec{x}) + \eta$ *satisfies* (ε, δ)*-differential privacy.*

▶ **Lemma 23** (Binomial Distribution is Smooth, [13]). *For any positive integer* n*,* $\gamma \in [0, 1/2]$*,* $\alpha \in [0, 1]$*, and any* $k \leq \alpha\gamma n/2$*, the distribution* $\mathrm{Bin}(n, \gamma)$ *is* (ε, δ, k)*-smooth with*

$$\varepsilon = \ln \frac{1+\alpha}{1-\alpha} \qquad \text{and} \qquad \delta = \exp\left(-\frac{\alpha^2\gamma n}{8}\right) + \exp\left(-\frac{\alpha^2\gamma n}{8+2\alpha}\right).$$

▶ **Corollary 24.** *Fix any $\varepsilon, \delta \in [0,1]$. Let $n \geq (100/\varepsilon^2) \cdot \ln(2/\delta)$. The algorithm $\mathcal{M}_{neg}$ that takes as input $\vec{x} \in \{0,-1\}^n$ then samples*

$$\eta \sim \mathrm{Bin}\left(n,\, 50 \cdot \frac{\ln(2/\delta)}{n\varepsilon^2}\right)$$

and reports $\eta + \sum x_i$ satisfies (ε, δ)-differential privacy.

Proof. When $\alpha = (e^\varepsilon - 1)/(e^\varepsilon + 1)$ observe that $\alpha \in [\varepsilon/\sqrt{5}, 1)$ and Lemma 23 implies that η is sampled from an $(\varepsilon, \delta, 1)$-smooth distribution:

$$\ln \frac{1+\alpha}{1-\alpha} = \ln \frac{(e^\varepsilon + 1) + (e^\varepsilon - 1)}{(e^\varepsilon + 1) - (e^\varepsilon - 1)} = \varepsilon$$

and

$$\begin{aligned} \exp\left(-\frac{\alpha^2\gamma n}{8}\right) + \exp\left(-\frac{\alpha^2\gamma n}{8+2\alpha}\right) &\leq 2\exp\left(-\frac{\alpha^2\gamma n}{10}\right) && (\alpha < 1) \\ &\leq 2\exp\left(-\frac{\gamma\varepsilon^2 n}{50}\right) \\ &= \delta. \end{aligned}$$

So by Lemma 22, we have $\mathcal{M}_{\mathrm{neg}}$ is (ε, δ)-differentially private. ◀