Private Counting from Anonymous Messages: Near-Optimal Accuracy with Vanishing Communication Overhead

Badih Ghazi [1] **Ravi Kumar** [1] **Pasin Manurangsi** [1] **Rasmus Pagh** [2 1]

Abstract

Differential privacy (DP) is a formal notion for quantifying the privacy loss of algorithms. Algorithms in the central model of DP achieve high accuracy but make the strongest trust assumptions whereas those in the local DP model make the weakest trust assumptions but incur substantial accuracy loss. The shuffled DP model (Bittau et al., 2017; Erlingsson et al., 2019; Cheu et al., 2019) has recently emerged as a feasible middle ground between the central and local models, providing stronger trust assumptions than the former while promising higher accuracies than the latter. In this paper, we obtain practical communication-efficient algorithms in the shuffled DP model for two basic aggregation primitives used in machine learning: 1) binary summation, and 2) histograms over a moderate number of buckets. Our algorithms achieve accuracy that is arbitrarily close to that of central DP algorithms with an expected communication per user essentially matching what is needed without any privacy constraints! We demonstrate the practicality of our algorithms by experimentally comparing their performance to several widely-used protocols such as Randomized Response (Warner, 1965) and RAPPOR (Erlingsson et al., 2014).

1. Introduction

Motivated by the need for scalable, distributed privacy-preserving machine learning, there has been an intense interest, both in academia and industry, on designing algorithms with low communication overhead and high accuracy while protecting potentially sensitive, user-specific information. While many notions of privacy have been proposed, *differential privacy* (DP) (Dwork et al., 2006b;a) has become by far the most popular and well-studied candidate, leading to several real-world deployments at companies such as Google (Erlingsson et al., 2014; Shankland, 2014), Apple (Greenberg, 2016; Apple Differential Privacy Team, 2017), and Microsoft (Ding et al., 2017), and in government agencies such as the U.S. Census Bureau (Abowd, 2018). Most research has focused on the *central* model of DP where a curator, who sees the raw user data, is required to release a private data structure. While many accurate DP algorithms have been discovered in this framework, the requirement that the curator observes the raw data constitutes a significant obstacle to deployment in many industrial settings where the users do not necessarily trust the central authority. To circumvent this limitation, several works have studied the *local* model of DP (Kasiviswanathan et al., 2008) (also (Warner, 1965)), which enforces the more stringent constraint that each message sent from a user device to the server is private. While requiring near-minimal trust assumptions, the local model turns out to inherently suffer from large estimation errors. For numerous basic tasks, including binary summation and histograms that we study in this work, errors are at least on the order of $\sqrt{n}$, where n is the number of users (Beimel et al., 2008; Chan et al., 2012).

Shuffled Privacy Model. The shuffled (aka. anonymous) model of privacy has recently generated significant interest as a potential compromise between the central and local frameworks: having trust assumptions better than the former but enabling estimation accuracies higher than the latter. While the shuffled model was originally studied in the field of cryptography by Ishai et al. (2006) in their work on cryptography from anonymity, it was first suggested as a framework for privacy-preserving computations by Bittau et al. (2017) in their Encode-Shuffle-Analyze architecture. This setting only requires the multiset of *anonymized* messages that are transmitted by the different users to be private. Equivalently, this corresponds to the setup where a trusted shuffler *randomly permutes* all incoming messages from the users before passing them to the analyzer. This is illustrated in Figure 1. We point out that several efficient cryptographic implementations of the shuffler have been considered includ-

[*]Equal contribution [1]Google Research, Mountain View [2]IT University of Copenhagen. Correspondence to: Badih Ghazi <badihghazi@gmail.com>, Ravi Kumar <ravi.k53@gmail.com>, Pasin Manurangsi <pasin@google.com>, Rasmus Pagh <pagh@itu.dk>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

ing mixnets, onion routing, secure hardware, and third-party servers (see, e.g., (Ishai et al., 2006; Bittau et al., 2017) for more details). As in all previous work on the shuffled model, we treat the shuffler as a black box.

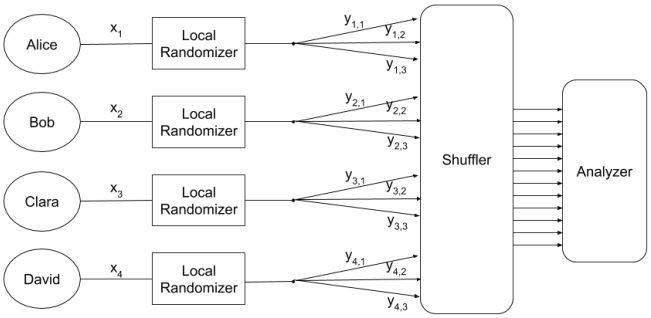

Figure 1. The Shuffled Model.

DP in the shuffled model was first formally investigated, independently, by Erlingsson et al. (2019) and Cheu et al. (2019). Several recent works have aimed to determine the optimal trade-offs between communication, accuracy, and privacy in this model for various algorithmic tasks (Balle et al., 2019; Ghazi et al., 2019b;a; 2020b; Balle et al., 2020; Balcer & Cheu, 2020).

Summation. One of the most basic distributed computation problems is *summation* (aka. aggregation) where the goal of the analyzer is to estimate the sum of the user inputs. In machine learning, and specifically in the nascent field of federated learning (Konečnỳ et al., 2016) (see, e.g., (Kairouz et al., 2019) for a recent survey), private summation enables private Stochastic Gradient Descent (SGD), which in turn allows the private training of deep neural networks that are guaranteed not to overfit to any user-specific information. Moreover, summation is perhaps the most primitive functionality in database systems in general, and in private implementations in particular (see, e.g., (Kotsogiannis et al., 2019; Wilson et al., 2019; Suresh et al., 2017)).

A notable special case is *binary* summation (aka. *counting query*) where each user holds a bit as an input and the goal of the analyzer is to estimate the number of users whose input equals 1. The vector version of this problem captures, e.g., the case where gradients have been quantized to bits in order to reduce the communication cost (e.g., the 1-bit SGD of Seide et al. (2014)). As observed in Blum et al. (2005), binary summation is of particular interest in ML since it is sufficient for implementing any learning algorithm based on *statistical queries* (Kearns, 1998), which includes most of the known PAC-learning algorithms.

Several recent works have studied private summation in the shuffled model (Cheu et al., 2019; Balle et al., 2019; Ghazi et al., 2019b; Balle et al., 2020; Ghazi et al., 2020b;a). These results achieve DP with parameters ε, δ (defined in Section 2). Cheu et al. (2019) showed that the standard Randomized Response (which goes back to Warner (1965) in the local DP case) is (ε, δ)-DP and incurs a squared error of $O(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta})$ with high probability. All mentioned works have also studied *real* summation, culminating in an (ε, δ)-DP protocol in the shuffled model with error arbitrarily close to a discrete Laplace random variable with parameter $1/\varepsilon$, and where each user sends $O(1 + \frac{\log(1/\delta)}{\log n})$ messages of $O(\log n)$ bits each (Ghazi et al., 2020b; Balle et al., 2020).

Histograms. A generalization of the binary summation problem is that of computing *histograms* (aka. *frequency oracles* or *frequency estimation*), where each user holds an element from some finite set $[B] := \{1, \ldots, B\}$ and the goal of the analyzer is to estimate for all $j \in [B]$, the number of users holding element j as input. Computing histograms is fundamental in data analytics and is well-studied in DP (e.g., (Kairouz et al., 2016; Acharya & Sun, 2019; Suresh, 2019)), as private histogram procedures can be used as a black-box to solve important algorithmic problems such as *heavy hitters* (e.g., (Bassily et al., 2017)) as well as unsupervised machine learning tasks such as *clustering* (e.g., (Stemmer, 2020)); furthermore, computing histogram is intimately related to *distribution estimation* (e.g., (Kairouz et al., 2016; Acharya & Sun, 2019)). In central DP, the smallest possible estimation error for histograms is known to be $\Theta(\min(\frac{\log(1/\delta)}{\varepsilon}, \frac{\log B}{\varepsilon}, n))$ (e.g., Section 7.1 in Vadhan (2017)). On the other hand, the smallest possible error in local DP is $\Theta(\min(\frac{\sqrt{n \log B}}{\varepsilon}, n))$ provided $\delta < 1/n$ (Bassily & Smith, 2015). In the shuffled DP setting and for ε a constant and δ inverse-polynomial in n, the tight estimation error for single-message protocols (where each user sends a single message) is $\widetilde{\Theta}(\min\{n^{1/4}, \sqrt{B}\})$, whereas multi-message protocols with both error and per-user communication that are logarithmic in B and n are known (Ghazi et al., 2019a; Erlingsson et al., 2020). Recently, Balcer & Cheu (2020) obtained a protocol with error independent of B but logarithmic in n, albeit with a per-user communication of $O(B)$ messages each consisting of $O(\log B)$ bits.

Two recent works (Wang et al., 2019; Erlingsson et al., 2020) studied private histograms in extensions of the shuffled model to multiple shufflers. Wang et al. (2019) uses Randomized Response whereas Erlingsson et al. (2020) uses a fragmented version of RAPPOR (Erlingsson et al., 2014).

In this work we focus on the regime where $B \ll n$, which captures numerous practical scenarios since the number of buckets is typically small compared to the population size.

1.1. Main Results

For the binary summation problem, we give the first private protocol in the shuffled model achieving mean squared er-

ror (MSE) arbitrarily close to the central performance of the Discrete Laplace mechanism while having an *expected* communication per user of $1 + o(1)$ messages of 1 bit each.

Theorem 1 (Binary Summation Protocol). *For every $\varepsilon \leq O(1)$ and every $\delta, \gamma \in (0, 1/2)$, there is an (ε, δ)-DP protocol for binary summation in the multi-message shuffled model with error equal to a Discrete Laplace random variable with parameter $(1-\gamma)\varepsilon$ and with an expected communication per user of $1 + O\left(\frac{\log^2(1/\delta)}{\gamma\varepsilon^2 n}\right)$ bits.*

We extend Theorem 1 to a protocol for histograms that, with a moderate number of buckets, has error arbitrarily close to the central DP performance of the Discrete Laplace mechanism while using essentially minimal communication.

Corollary 2 (Histogram Protocol). *For every $\varepsilon \leq O(1)$ and every $\delta, \gamma \in (0, 1/2)$, there is an (ε, δ)-DP protocol for histograms on sets of size B in the multi-message shuffled model, with error equal to a vector of independent Discrete Laplace random variables each with parameter $\frac{(1-\gamma)\varepsilon}{2}$ and with an expected number of messages sent per user equal to $1 + O\left(\frac{B\log^2(1/\delta)}{\gamma\varepsilon^2 n}\right)$, each consisting of $\lceil \log B \rceil + 1$ bits.*

For the standard setting of constant ε and δ inverse-polynomial in n and for an arbitrarily small positive constant γ, the expected communication per user in Theorem 1 is $1 + o(1)$ bits. Note that 1 bit of communication per user is required for accurate estimation of the binary summation even in the absence of any privacy constraints. Likewise, the expected communication per user in Corollary 2 is $\lceil \log B \rceil + 1 + o(1)$ bits. Here again, $\log B$ bits of communication per user is required for accurate estimation of the histogram even in the absence of any privacy constraints.

A natural question in the context of Theorem 1 and Corollary 2 is whether the same accuracy and communication can be achieved by a *single-message* protocol in the shuffled model. For histograms, this is impossible given the $\tilde{\Omega}(\min\{n^{1/4}, \sqrt{B}\})$ lower bound of Ghazi et al. (2019a) on the ℓ_∞-error of any single-message protocol whereas the expected ℓ_∞ error in Corollary 2 is at most $O(\frac{\log B}{\varepsilon})$. We prove that this is also impossible for binary summation:

Theorem 3 (Binary Summation Lower Bound). *Let $\delta = 1/n^{\Omega(1)}$ and $\varepsilon \leq O(1)$. Then, any (ε, δ)-DP protocol for Binary Summation in the single-message shuffled model should incur an expected squared error of at least $\Omega(\log n)$.*

In light of the lower bound in Theorem 3 and the aforementioned lower bound of Ghazi et al. (2019a), it is striking that the protocols in Theorem 1 and Corollary 2 can get arbitrarily close to the central performance of the Discrete Laplace mechanism while being *almost* single-message: the vast majority of users send a single message (consisting of a single bit in the binary summation protocol and $\lceil \log B \rceil + 1$ bits in the histogram protocol) while only a random $o(1)$ fraction of users sends more than one message!

We point out that, as in previous work in the shuffled and local models of DP, the communication costs in Theorem 1 and Corollary 2 exclude the encryption costs. However, as different messages sent by the users have to be encrypted separately, the encryption overhead increases with the number of messages, and hence our almost single-message protocols would be even more appealing compared to other multi-message procedures as in Ghazi et al. (2020b); Balle et al. (2020); Ghazi et al. (2019a) when the encryption costs are taken into account.

Experimental Evaluation. We implement our algorithms and compare their performance to several alternatives proposed in the literature, both for binary summation and histogram. For the latter, we evaluate the algorithms on public 1940 US Census IPUMS dataset, considering both categorical and numerical features. Our experiments support our theoretical analysis: for a broad setting of n, ε, δ, and B, we incur small communication overhead while achieving near-central errors that are noticeably smaller than previous protocols. The experimental results are presented in the appendix.

Remark 4. *We note that our algorithms in Theorem 1 and Corollary 2 can be used to learn the empirical distribution of the users' data up a small error. In light of the near-optimality properties of the Discrete Laplace distribution in the central DP model (Ghosh et al., 2012), our algorithms are also close to optimal. This holds for general error measures including the ℓ_1, ℓ_2, and ℓ_∞ norms, which are well-studied in the literature on distribution estimation and learning (e.g., (Kairouz et al., 2016; Acharya & Sun, 2019)).*

1.2. Overview of Techniques

Before outlining the proof of Theorem 1, we first note that the Discrete Laplace mechanism in the central model incurs only a *constant* MSE. To get a similar bound in the shuffled model, any single-message protocol—in particular, Randomized Response and RAPPOR (Erlingsson et al., 2014) (which for binary summation coincides with Randomized Response)—is ruled out by Theorem 3. Furthermore, the recent histogram protocols of Ghazi et al. (2019a); Erlingsson et al. (2020), are also not applicable since they all incur an MSE of $\Omega(\log n)$. Theorem 1 also guarantees vanishing communication overhead: this rules out the split-and-mix protocol in Ghazi et al. (2020b); Balle et al. (2020) and a recent protocol of Ghazi et al. (2020a).

We next recall the prototypical private binary summation procedures in the central setup. If user i's input is x_i, then the analyzer simply computes the correct sum $\sum_{i\in[n]} x_i$ and then adds to it a random variable sampled from some

probability distribution $\mathcal{D}$. A common choice of $\mathcal{D}$ is the Discrete Laplace distribution with parameter ε, which yields an $(\varepsilon, 0)$-DP protocol for binary summation with an asymptotically tight MSE of $O(1/\varepsilon^2)$; this is known to be optimal in the central DP model (Ghosh et al., 2012).

Using Infinitely Divisible Distributions. In order to emulate the prototypical central model mechanism in the shuffled model, we need to distribute both the signal and the noise over the n users. Distributing the signal can be naturally done by having each user i merely send their true input bit x_i. Distributing the noise is significantly more challenging since the shuffled model is symmetric and does not allow coordination of noise across users. Consider the framework, captured in Algorithms 1 and 2 on page 5, where (a) each user sends (possibly several) bits to the shuffler and (b) the analyzer counts the number of 1s received from the shuffler and outputs it as a proxy for the true sum (possibly after subtracting a fixed bias term). In this case, we would need to decompose the noise random variable into n i.i.d. non-negative components, and have each user sample and transmit one component in unary. Distributions that are decomposable into the sum of n i.i.d. (not necessarily non-negative) samples for any positive integer n are well-studied in probability theory and are known as *infinitely divisible*. In DP, the Discrete Laplace distribution was observed to be infinitely divisible by Goryczka & Xiong (2015), and this property was used by Balle et al. (2020) for real summation in the shuffled model, albeit with several messages per user, each consisting of $\Omega(\log n)$ bits. However, decomposing the Discrete Laplace distribution into a sum of i.i.d. *non-negative* samples—as required by our template above—is clearly impossible since its support contains negative values.

One basic discrete non-negative infinitely divisible distribution is the *Poisson* distribution with parameter λ, which can be sampled by summing n i.i.d. samples from a Poisson distribution with parameter λ/n, for any positive integer n. The resulting *Poisson mechanism* can thus be used as a candidate binary summation procedure in the shuffled model. It turns out that this mechanism is (ε, δ)-DP if we set λ to $O\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)$ (see Theorem 11). In this case, the expected communication cost of transmitting the per-user noise is equal to the expectation λ/n, which is much smaller than 1. We note that, for a reason explained in Section 3.2, we can further reduce the communication by considering the Negative Binomial distribution $\mathrm{NB}(r, p)$. This distribution is infinitely divisible as a random sample from $\mathrm{NB}(r, p)$ can be generated by summing n i.i.d. samples from $\mathrm{NB}(r/n, p)$, for any positive integer n.

Unfortunately, it turns out we cannot hope to achieve near-central accuracy using any *non-negative* infinitely divisible distribution. Specifically, we prove in Section 3.3 that for every such noise distribution, the incurred MSE will grow asymptotically with $\log(1/\delta)$. This is in sharp contrast with the error in the central model, which is independent of δ.

Unary Encoding and Correlated Noise. Instead, the support of our noise distribution has to also contain negative values. To allow this, a natural extension of the above template algorithm is to let each message consists of either an increment (e.g., $+1$) value or a decrement (e.g., -1) value. This template leads to a distributed noise strategy that can achieve near-central accuracy, described next. We know from Goryczka & Xiong (2015) that the Discrete Laplace distribution with parameter ε is the same as the distribution of the difference of two independent $\mathrm{NB}(1, e^{-\varepsilon})$ random variables, and is thus infinitely divisible. This noise can be distributed in the shuffled model by letting each user sample two independent random variables Z^1 and Z^2 from $\mathrm{NB}(1/n, e^{-\varepsilon})$, and send Z^1 increment messages and Z^2 decrement messages to the shuffler. This mechanism would achieve the same error as the central Discrete Laplace mechanism. However, since the analyzer can still see the number of increment messages, this scheme is no more private than the (non-negative) mechanism with noise distribution $\mathrm{NB}(1, e^{-\varepsilon})$, and thus cannot be (ε, δ)-DP by virtue of the lower bound (Section 3.3).

To leverage the power of sending both positive and negative messages, we correlate the input-dependent and noise components sent by the users so that the analyzer is unable to extract much information about the user inputs from one type of messages. We do so by employing a *unary version* of the split-and-mix procedure of Ishai et al. (2006); Ghazi et al. (2020b); Balle et al. (2020). Namely, in addition to the aforementioned random variables Z^1 and Z^2, each user will independently sample a third random variable Z^3 from another infinitely divisible distribution, and will send $Z^1 + Z^3$ increment messages and $Z^2 + Z^3$ decrement messages (see Algorithm 3 on page 7). Note that in this case, when Z^3 is sufficiently "spread out", the analyzer cannot extract much information from counting the number of increments alone, since the noise from Z^3 already overwhelms the user inputs. We formalize this intuition by proving that, for carefully selected infinitely divisible noise distributions, the resulting mechanism is (ε, δ)-DP and incurs an error that can be made arbitrarily close to that of the central Discrete Laplace mechanism, while incurring an expected communication overhead per user that goes to 0 with as n increases.

To prove Corollary 2, we run the binary summation protocol in parallel on all B buckets, and instead of sending ± 1 valued messages, we concatenate each with the length $\lceil \log B \rceil$ binary expansion of the index of the bucket being incremented/decremented. While a straightforward implementation of the randomizer has a running time of $\Omega(B)$, we show, using the characterization of infinitely divisible distributions in terms of Discrete Compound Poisson (DCP) distributions,

that the expected running time can be significantly reduced to the order of the expected per-user communication cost.

Size-Freeness. Infinitely divisible noise mechanisms are much easier to deploy in practice compared to general schemes. This is because for fixed (ε, δ), the "noise parameter" of infinitely divisible mechanisms is independent of the number of users n: e.g., in the case of the Poisson Mechanism, the parameter λ only depends on ε and δ. In contrast, computing near-optimal parameters in the shuffled model of the noise parameters for non-infinitely divisible schemes such as Randomized Response (Warner, 1965) and RAPPOR (Erlingsson et al., 2014) requires re-running a time-expensive algorithm for each new value of n (see the appendix for more details). This can be undesirable in practice, especially for real-time applications.

1.3. Organization

We provide some background and notation in Section 2. In Sections 3 and 4 we prove Theorem 1. Our experimental setup and results are presented in Section 5. We conclude with some open questions in Section 6. Corollary 2 and Theorem 3 are proved in Appendices C and D.

2. Preliminaries

Let n be the number of users and $[n] := \{1, \ldots, n\}$. In this work, we only consider discrete probability distributions that are supported on (possibly negative) integers. We write $X \sim \mathcal{D}$ to denote a random variable X sampled according to $\mathcal{D}$. We let $\mathrm{supp}(\mathcal{D})$ denote the support of $\mathcal{D}$, $\mathbb{E}[\mathcal{D}]$ its mean, and $\mathrm{Var}(\mathcal{D})$ its variance. For two distributions $\mathcal{D}$ and $\mathcal{D}'$, we denote by $\mathcal{D} + \mathcal{D}'$ the distribution of $X + X'$ where X and X' are independently sampled from $\mathcal{D}$ and $\mathcal{D}'$ respectively; $\mathcal{D} - \mathcal{D}'$ is defined similarly. For integer k, we denote by $k + \mathcal{D}$ the distribution of $k + X$ when $X \sim \mathcal{D}$. For any $x \in \mathbb{Z}$ we let $\mathcal{D}(x)$ denote $\Pr_{Z \sim \mathcal{D}}[Z = x]$. We denote by $\mathbf{x} \sim \mathbf{x}'$ two datasets (i.e., n-dimensional vectors) $\mathbf{x}$ and $\mathbf{x}'$ differing on a single user's data (i.e., a single coordinate).

Definition 5 (Differential Privacy (Dwork et al., 2006a;b))**.** *For any parameters $\varepsilon \geq 0$ and $\delta \in [0, 1]$, a randomized mechanism $\mathcal{M}$ is* (ε, δ)-differentially private (DP) *if for every pair of $\mathbf{x} \sim \mathbf{x}'$ and for every subset $\mathcal{S}$ of transcripts of $\mathcal{M}$, it holds that $\Pr[\mathcal{M}(\mathbf{x}) \in \mathcal{S}] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(\mathbf{x}') \in \mathcal{S}] + \delta$, where the probabilities are over the randomness in $\mathcal{M}$.*

Shuffled Model. A protocol in the shuffled privacy model consists of three procedures: a *local randomizer* that sends one or several messages depending on its input, a *shuffler* that randomly permutes all incoming messages, and an *analyzer* that takes in the output of the shuffler and returns the final output of the protocol. Privacy is required to be guaranteed with respect to the output of the shuffler.

Algorithm 1 $\mathcal{D}$-Distributed Randomizer.

procedure $\textsc{Randomizer}_{\mathcal{D},n}(x)$
  Sample $Z \sim \mathcal{D}_{/n}$
  Send $x + Z$ messages, where each message is 1

3. The $\mathcal{D}$-Distributed Mechanisms

In this section, we propose and study a family of simple mechanisms in the shuffled model for the binary summation problem. While the mechanisms in this section do not achieve the accuracy promised in Theorem 1, they will serve as an important building block to our eventual algorithm in Section 4. In fact, we will need the privacy guarantee of these mechanisms against a generalization of bit summation called Δ-summation defined as follows. For $\Delta \in \mathbb{N}$, in the Δ-*summation* task, the input to each user is a number x_i in $\{0, \ldots, \Delta\}$ and the goal is to compute $\sum_{i \in [n]} x_i$. When $\Delta = 1$, this task is the same as binary summation.

To define our protocol, we first recall that a standard strategy for achieving DP in the central model is to simply add noise to the correct answer. We will refer to such a mechanism the $\mathcal{D}$ *Mechanism* when the noise distribution is $\mathcal{D}$.

Definition 6. *For any distribution $\mathcal{D}$, the* $\mathcal{D}$ Mechanism *for computing a function $f : \mathcal{X}^n \to \mathbb{Z}^d$ is defined as the mechanism that, on input $\mathbf{x} \in \mathcal{X}^n$, outputs $f(\mathbf{x}) + (Y_1, \ldots, Y_d)$ where $Y_i \sim \mathcal{D}, i \in [d]$ are independent.*

The definition of the $\mathcal{D}$ Mechanism applies even when $\mathrm{supp}(\mathcal{D})$ contains negative integers, but in this section we focus on distributions $\mathcal{D}$ that are supported on non-negative integers and that are infinitely divisible as defined next.

Definition 7 (Infinite Divisibility)**.** *A distribution $\mathcal{D}$ is said to be* infinitely divisible *(abbreviated* ∞-div*) if for every $n \in \mathbb{N}$, there exists a distribution $\mathcal{D}_{/n}$ such that $(X_1 + \cdots + X_n) \sim \mathcal{D}$ where $X_i \sim \mathcal{D}_{/n}, i \in [d]$ are independent.*

For an ∞-div $\mathcal{D}$ on non-negative integers, we define the $\mathcal{D}$*-Distributed Mechanism* in the shuffled model as follows:

Algorithm 2 $\mathcal{D}$-Distributed Analyzer.

procedure $\textsc{Analyzer}_{\mathcal{D}}$
  $U \leftarrow$ number of messages received
  return $U - \mathbb{E}[\mathcal{D}]$

Privacy. Observe that, from the analyzer's perspective, it only sees U (because all the messages are identical) and U is distributed exactly as $\sum_{i \in [n]} x_i + \mathcal{D}$ by ∞-div of $\mathcal{D}$. From this, we immediately get that the privacy guarantee of the $\mathcal{D}$-Distributed Mechanism in the *shuffled* model is the same as that of the $\mathcal{D}$ Mechanism in the *central* model.

Observation 8. *For any $\varepsilon > 0$ and $\delta \in (0, 1)$, the $\mathcal{D}$-Distributed Mechanism is (ε, δ)-DP in the shuffled model*

for Δ-summation if and only if the $\mathcal{D}$ Mechanism is (ε, δ)-DP in the central model for Δ-summation.

Accuracy. As discussed above, we are guaranteed in Algorithm 2 that $U = \sum_{i\in[n]} x_i + \mathcal{D}$, which gives:

Observation 9. *The error of the $\mathcal{D}$-distributed Mechanism is distributed as $\mathcal{D} - \mathbb{E}[\mathcal{D}]$.*

Expected Communication. The expected number of messages sent by each user in Algorithm 1 is $x + \mathbb{E}[\mathcal{D}_{/n}] = x + \frac{\mathbb{E}[\mathcal{D}]}{n}$. Using the fact that $x \in \{0, \ldots, \Delta\}$, we get:

Observation 10. *The expected number of messages sent by a user in the $\mathcal{D}$-Distributed Mechanism is at most $\Delta + \frac{\mathbb{E}[\mathcal{D}]}{n}$.*

3.1. Example I: The Poisson Mechanism

Arguably, the simplest protocol in the family of $\mathcal{D}$-Distributed Mechanisms is the *Poisson Mechanism* that uses the Poisson distribution[1], which is ∞-div. In this protocol, $\mathcal{D}$ is $\mathrm{Poi}(\lambda)$ for some $\lambda \in \mathbb{R}^+$ and $\mathcal{D}_{/n}$ is simply $\mathrm{Poi}(\lambda/n)$. We now compute its privacy guarantee.

Theorem 11. *For any $\varepsilon > 0$, $\delta \in (0,1)$, and $\Delta \in \mathbb{N}$, the $\mathrm{Poi}(\lambda)$ Mechanism with $\lambda = \frac{16\log(10/\delta)}{(1-e^{-\varepsilon/\Delta})^2} + \frac{2\Delta}{1-e^{-\varepsilon/\Delta}}$, is (ε, δ)-DP in the central model for Δ-summation.*

By setting $\Delta = 1$ and using Observations 8, 9, and 10, we get the following for binary summation.

Corollary 12. *For any $0 < \varepsilon \le O(1)$ and $\delta \in (0,1)$, let λ be as in Theorem 11 with $\Delta = 1$. The $\mathrm{Poi}(\lambda)$-Distributed Mechanism is (ε, δ)-DP for binary summation in the shuffled model, each user sends at most $1 + O\left(\frac{\log(1/\delta)}{\varepsilon^2 n}\right)$ one-bit messages in expectation, and the MSE is $O\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)$.*

3.2. Example II: The Negative Binomial Mechanism

A disadvantage of the Poisson Mechanism is that, in the most important regime where $\frac{\varepsilon}{\Delta} \ll 1$, the expected number of messages sent is $1 + O\left(\frac{\log(1/\delta)}{n} \cdot \left(\frac{\Delta}{\varepsilon}\right)^2\right)$. In this subsection, we show how to reduce the dependency on $\frac{\Delta}{\varepsilon}$ from $(\frac{\Delta}{\varepsilon})^2$ to $\frac{\Delta}{\varepsilon}$, while retaining a similar error bound. (As we will see in Section 4, this dependency will also permeate to our eventual algorithm in the proof of Theorem 1.) Before doing so, we note that the $(\frac{\Delta}{\varepsilon})^2$ dependency is necessary for the Poisson Mechanism since it is well-known[2] that the MSE of any central $(\varepsilon, o(1))$-DP protocol for Δ-summation has to be at least $\Omega((\frac{\Delta}{\varepsilon})^2)$ and since the MSE of the $\mathrm{Poi}(\lambda)$ Mechanism is exactly λ, we must hence set $\lambda \ge (\frac{\Delta}{\varepsilon})^2$. Thus, the expected number of messages sent per user in the $\mathrm{Poi}(\lambda)$ Mechanism must be $1 + \frac{\lambda}{n} \ge 1 + \Omega\left(\frac{1}{n} \cdot \left(\frac{\Delta}{\varepsilon}\right)^2\right)$.

To circumvent this, we first observe that the above argument holds only because the parameter λ of the Poisson Mechanism governs both the MSE (i.e., the variance of the distribution) and the number of messages sent (i.e., the expectation of the distribution). This motivates us to seek an ∞-div distribution on the negative integers whose variance and mean can be very different. We consider the negative binomial distribution[3] $\mathrm{NB}(r,p)$ which is ∞-div as $\mathrm{NB}(r,p) = \sum_{i=1}^{n} \mathrm{NB}(\frac{r}{n}, p)$ for every $n \in \mathbb{N}$. Moreover, $\mathbb{E}[\mathrm{NB}(r,p)] = \frac{pr}{(1-p)}$ and $\mathrm{Var}[\mathrm{NB}(r,p)] = \frac{pr}{(1-p)^2}$, which can be very different when p is close to 1. We next show a DP guarantee for this Negative Binomial Mechanism.

Theorem 13. *For any $\varepsilon > 0, \delta \in [0,1)$, and $\Delta \in \mathbb{N}$, let $p = e^{-0.1\varepsilon/\Delta}$ and $r = 50 \cdot e^{\varepsilon/\Delta} \cdot \log\left(\frac{1}{\delta}\right)$. The $\mathrm{NB}(r,p)$ Mechanism is (ε, δ)-DP in the central model.*

Plugging $\Delta = 1$, we get the following corollary. When compared to Poisson Mechanism (Corollary 12), we achieve the same error bound but with a $\frac{1}{\varepsilon}$ instead of $\frac{1}{\varepsilon^2}$ multiplicative term in the expected number of additional messages sent.

Corollary 14. *For any $\varepsilon, \delta > 0$ with $\varepsilon \le O(1)$, let p, r be as in Theorem 13 with $\Delta = 1$. The $\mathrm{NB}(r,p)$-Distributed Mechanism is (ε, δ)-DP for binary summation in the shuffled model, each user sends at most $1 + O\left(\frac{\log(1/\delta)}{\varepsilon n}\right)$ one-bit messages in expectation, and the MSE is $O\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)$.*

3.3. A Lower Bound for $\mathcal{D}$-Distributed Mechanisms

The downside of the Poisson and Negative Binomial Mechanisms is that they suffer from an MSE of $O_\varepsilon(\log(\frac{1}{\delta}))$ instead of the $O(\frac{1}{\varepsilon^2})$ MSE, independent of δ, of the central Discrete Laplace Mechanism. It turns out that this dependency on $\log(\frac{1}{\delta})$ is necessary for every $\mathcal{D}$-Distributed Mechanism:

Lemma 15. *For any infinitely divisible distribution $\mathcal{D}$ on non-negative integers, if $\mathcal{D}$-Distributed Mechanism is (ε, δ)-DP in the shuffled model for binary summation, then the MSE of the mechanism is $\Omega_\varepsilon(\log(1/\delta))$.*

In other words, $\mathcal{D}$-Distributed Mechanisms do not suffice for the goal of achieving near-central error guarantees.

4. Correlated Distributed Mechanisms

We next present a family of protocols in the shuffled model that will overcome the lower bound barrier of Lemma 15 and achieve an accuracy/privacy trade-off arbitrarily close to the central model. We start by outlining the intuition behind the protocol. First, suppose hypothetically that we could somehow implement the $\mathcal{D}$ Mechanism in the shuffled

[1] $\mathrm{Poi}(\lambda)$ is defined as $\mathrm{Poi}(k;\lambda) = \lambda^k e^{-k}/k!$.

[2] This follows from the sensitivity of Δ-summation; see, e.g.. (Vadhan, 2017).

[3] $\mathrm{NB}(r,p)$ is defined as $\mathrm{NB}(k;r,p) = \binom{k+r-1}{k}(1-p)^r p^k$. We remark that the negative binomial distribution may be viewed as a generalization of the Poisson distribution: when taking $r \to \infty$ and letting $p = \lambda/r$, $\mathrm{NB}(r,p)$ point-wise converges to $\mathrm{Poi}(\lambda)$.

model when $\mathrm{supp}(\mathcal{D})$ can contain negative integers. Then, we would actually be done! This is because the Discrete Laplace distribution[4] is ∞-div as $\mathrm{DLap}(\varepsilon) = \mathrm{NB}(1, e^{-\varepsilon}) - \mathrm{NB}(1, e^{-\varepsilon})$ (see, e.g., (Kotz et al., 2001)), and $\mathrm{NB}(1, e^{-\varepsilon})$ is ∞-div, and is in fact the geometric distribution. Of course, the problem is that if we sample the number of messages from $\mathrm{DLap}(\varepsilon)_{/n}$ we will often try to send a *negative* number of messages, which is meaningless!

This leads us to using two types of messages: one for increments (denoted $+1$) and one for decrements (denoted -1). The $+1$ messages are sampled as in the $\mathrm{NB}(1, e^{-\varepsilon})$ Mechanism, and so are the -1 messages except that each user pretends that their input is 0 in this case. The analyzer's answer is the difference between the number of $+1$ messages and the number of -1 messages it receives. The aforementioned fact that the difference of two $\mathrm{NB}(1, e^{-\varepsilon})$ random variables is $\mathrm{DLap}(\varepsilon)$ implies that this protocol has the same accuracy as the central Discrete Laplace Mechanism, as desired. However, this protocol is not (ε, δ)-DP in the shuffled model. To see this, note that the analyzer sees the number of $+1$ messages received, and hence the protocol is no more private than the $\mathrm{NB}(1, e^{-\varepsilon})$ Mechanism, which is not (ε, δ)-DP as explained by Lemma 15. To overcome this, we "mask" the numbers of $+1$ and -1 messages by sampling a random variable Z from $\mathcal{D}_{/n}$ for some other ∞-div $\mathcal{D}$ over non-negative integers, and additionally send Z increments (i.e., $+1$) and Z decrements (i.e., -1). Clearly, this does not affect the accuracy of the analyzer, but we will show that it improves privacy. For every choice of ∞-div $\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3$ on non-negative integers, we define the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-*Correlated Distributed Mechanism*:

Algorithm 3 $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Randomizer

procedure $\textsc{Randomizer}_{\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3, n}(x)$
  Sample $Z^1 \sim \mathcal{D}^1_{/n}$
  Sample $Z^2 \sim \mathcal{D}^2_{/n}$
  Sample $Z^3 \sim \mathcal{D}^3_{/n}$
  Send $x + Z^1 + Z^3$ many $+1$ messages.
  Send $Z^2 + Z^3$ many -1 messages.

Algorithm 4 $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Analyzer

procedure $\textsc{Analyzer}_{\mathcal{D}^1, \mathcal{D}^2}$
  $U_{+1} \leftarrow$ number of $+1$ messages received.
  $U_{-1} \leftarrow$ number of -1 messages received.
  return $U_{+1} - U_{-1} - \mathbb{E}[\mathcal{D}^1 - \mathcal{D}^2]$

Accuracy and Communication Complexity. The accuracy and communication complexity of the protocol can be derived as in the $\mathcal{D}$-Distributed Mechanism from Section 3:

Observation 16. *The error of $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Distributed Mechanism is distributed as $(\mathcal{D}^1 - \mathcal{D}^2) - \mathbb{E}[\mathcal{D}^1 - \mathcal{D}^2]$.*

Observation 17. *The expected number of messages sent by each user in the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Distributed Mechanism is at most $1 + \frac{\mathbb{E}[\mathcal{D}^1] + \mathbb{E}[\mathcal{D}^2] + 2 \cdot \mathbb{E}[\mathcal{D}^3]}{n}$.*

Privacy. The crux of our DP proof for the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Mechanism is the following theorem:

Theorem 18. *Let $\Delta > 0$ and $\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3$ satisfy:*

- *(Privacy of True Noise) The $(\mathcal{D}^1 - \mathcal{D}^2)$ Mechanism is ε_1-DP for binary summation in the central model.*
- *(Privacy of Correlated Noise) The $\mathcal{D}^3$ Mechanism is $(\varepsilon_2, \delta_2)$-DP for Δ-summation in the central model.*
- *(Concentration of Neg. Noise) $\Pr_{Y \sim \mathcal{D}^2}[Y > \Delta] \le \delta_3$.*

Then, the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Mechanism is $(\varepsilon_1 + \varepsilon_2, e^{\varepsilon_1} \cdot \delta_2 + 2e^{2\varepsilon_1} \cdot \delta_3)$-DP in the shuffled model.

Proof Overview. All the analyzer sees is $(U_{+1}, U_{-1}) = (\sum_{i \in [n]} x_i + \hat{Z}^1 + \hat{Z}^3, \hat{Z}^2 + \hat{Z}^3)$ where for $j \in [3]$, $\hat{Z}^j \sim \mathcal{D}^j$. Since there is bijection between (U_{+1}, U_{-1}) and $(U_{+1} - U_{-1}, U_{-1}) = (\sum_{i \in [n]} x_i + \hat{Z}^1 - \hat{Z}^2, \hat{Z}^2 + \hat{Z}^3)$, we may consider the distribution on the latter. The first coordinate is the same as the analyzer's view from the $(\mathcal{D}^1 - \mathcal{D}^2)$ Mechanism, which is ε_1-DP by the first assumption. Once we condition on $U_{+1} - U_{-1}$ being equal to some value, we are left to consider a distribution on U_{-1}. This distribution is not the same as the original one (before conditioning) since U_{-1} and $U_{+1} - U_{-1}$ are correlated. But this correlation only comes via $\hat{Z}^2$, as $\hat{Z}^3$ does not appear in $U_{+1} - U_{-1}$. By the concentration of $\mathcal{D}^2$ (the third assumption), $\hat{Z}^2$ is rarely larger than Δ. When $\hat{Z}^2 \le \Delta$, we can use the $(\varepsilon_2, \delta_2)$-DP of the $\mathcal{D}^3$ Mechanism for Δ-summation to argue the privacy of the protocol. The proof follows this intuition while carefully tracking the privacy loss in each step.

4.1. Near-Central Accuracy with Shuffled Mechanisms

We use Theorem 18 to derive our "near-central" protocol (Theorem 1). Specifically, we set $\mathcal{D}^1, \mathcal{D}^2$ so that $\mathcal{D}^1 - \mathcal{D}^2 = \mathrm{DLap}(0.99\varepsilon)$, which implies that the $(\mathcal{D}^1 - \mathcal{D}^2)$ Mechanism is 0.99ε-DP in the central model. The remaining privacy budget of 0.01ε is allocated to the $\mathcal{D}^3$ Mechanism, which we set to be the $\mathrm{NB}(\cdot, \cdot)$ Mechanism with appropriate parameters. We use the Negative Binomial rather than the Poisson distribution as the former (Theorem 13) has a smaller communication cost than the latter (Theorem 11).

[4] $\mathrm{DLap}(s)$ is defined as $\mathrm{DLap}(k; s) = \frac{1}{C(s)} \cdot e^{-|k| \cdot s}$, where $C(s) = \sum_{k=-\infty}^{\infty} e^{-|k| \cdot s}$ is the normalization constant.

5. Experimental Evaluation and Results

5.1. Binary Summation

In this section, we evaluate our protocols. Specifically, we consider the Poisson Distributed Mechanism (Theorem 11) and the Correlated Distributed Noise Mechanism (Theorems 1 and 18). We consider the root mean square error (RMSE), which is independent of the input data for all methods considered, and for this reason there is no need to consider performance on particular data sets. For each setting of ε, δ, our parameters are selected in an accurate manner; this is explained in detail in the Appendix F. We only note here that for the Correlated Distributed Mechanism, we set $\mathcal{D}^1 = \mathcal{D}^2 = \mathrm{NB}(1, e^{-\varepsilon_1})$ with ε_1 such that the RMSE of the protocol is 20% more than that of the (central) $\mathrm{DLap}(\varepsilon)$ Mechanism. We compare our algorithms against the classic *Randomized Response (RR)* algorithm, where a user with input x sends x w.p. p, and $1-x$ w.p. $1-p$, for some parameter $p \in [0, 1/2]$.

Error. We compute the errors as ε or δ varies. The corresponding plots are shown in Figure 2; we include the error of the (central) Discrete Laplace Mechanism for comparison (but of course this is not directly implementable in the shuffled model). We remark that the RMSEs of our protocols are independent of the number of users n, and only the RMSE of RR depends on n, which we choose to be 10,000. While the Correlated Distributed protocol has a constant RMSE as we vary δ, both Poisson and RR incur larger RMSEs. In particular, when $\delta = 10^{-6}, \varepsilon = 1$, the RMSE of the Correlated Distributed protocol is 3.5 times less than that of Poisson and RR (which essentially coincide). The fact that the Poisson Mechanism and RR have essentially the same RMSE should come as no surprise, since the binomial distribution $\mathrm{Bin}(n, p)$ converges (in the distributional sense) to the Poisson distribution $\mathrm{Poi}(np)$ as $n \to \infty$ and p is kept constant. This means that we would prefer RR in this case, since it always sends one message.

Communication Complexity. Figure 3 shows the plots of the expected number of additional messages sent by each user in our Poisson and Correlated Distributed Mechanisms. Here we let $n = 10,000$. In the reasonable setting where $\delta = 10^{-6}$ and $\varepsilon = 1$, the expected number of additional messages sent in the Correlated Distributed Mechanism is only 0.04, whereas in the Poisson Mechanism, it is even smaller at 0.0003. Even in the more extreme case of $\varepsilon = 0.1$, the former is still 0.278 and the latter is only 0.141.

5.2. Histograms

We have also performed experiments on the histogram versions of our Correlated Distributed and Poisson Mechanisms, and compare them against three algorithms from the literature: B-Randomized Response (B-RR), RAPPOR (Erlingsson et al., 2014), and Fragmented RAPPOR (Erlingsson et al., 2020). Each of these three can be viewed as a B-ary generalization of the binary RR. We ran these algorithms on two IPUMS datasets (Ruggles et al., 2019). Due to space constraints, the full description of the experiments and results are deferred to Appendix E. Here we just summarize our findings: For most parameters, RAPPOR incurs significantly larger errors than B-RR. Moreover, similarly to how RR mirrors the Poisson Mechanism in the binary case, Fragmented RAPPOR gives almost the same results as Poisson for histograms. In terms of RMSE, our correlated mechanism is significantly closer to the central model error than its competitors. The plots are in fact very similar to those of binary summation for the Correlated Distributed and Poisson Mechanisms, with RR doing 25-50% worse than Poisson. In terms of ℓ_∞, RR incurs more than $3\times$ ℓ_∞ errors compared to our algorithms. Furthermore, as corroborated by theory (Ghazi et al., 2019a), the error of RR grows quickly with the number B of buckets, while those of our mechanisms grow very slowly.

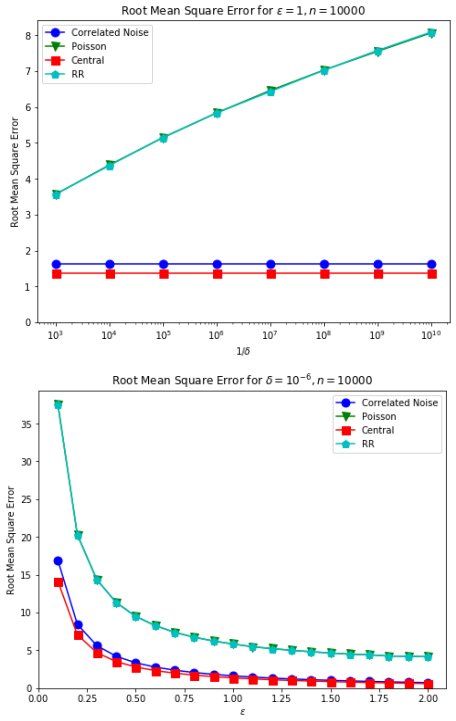

Figure 2. RMSE of the protocols for binary summation. Note that the RMSEs of RR and Poisson are essentially the same.

6. Conclusions and Open Questions

We proposed DP algorithms in the shuffled model for binary summation and histograms with accuracy arbitrarily close to the central model and a vanishing communication overhead. There are several questions left open by our work. One

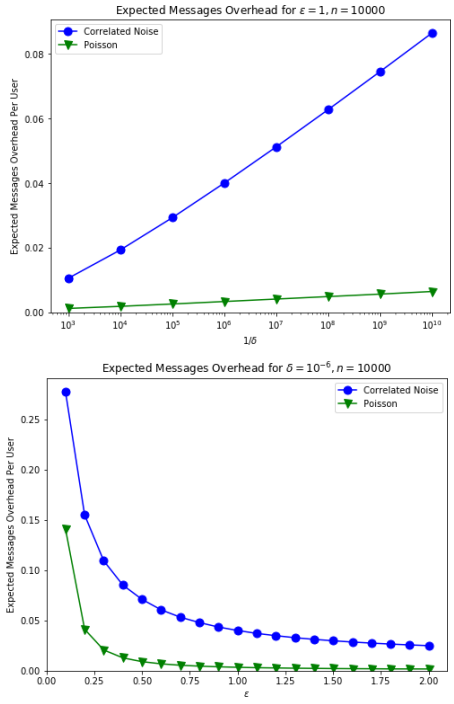

Figure 3. Expected additional number of messages sent by each user for binary summation. While we use $n = 10^4$, this expectation scales linearly in $1/n$. E.g., if $n = 10^5$, the plots will look the same, but with the y-axis scaled down by a factor of 10.

is to obtain algorithms achieving near-central performance with negligible communication overhead for the problems of real summation, vector summation, and histograms over a large number $B \geq \Omega(n)$ of buckets. Another question is to prove a lower bound against *expected single-message* protocols (that can send 0, 1 or more messages in the worst-case) as our lower bound in Theorem 3 does not hold in this case. A very interesting related direction is to build on our algorithms to improve the communication complexity of DP stochastic gradient descent in a federated learning setup.

Acknowledgements

We would like to thank Noah Golowich, Laura Peskin and Alex Zani for insightful discussions and feedback.

References

Abowd, J. M. The US Census Bureau adopts differential privacy. In *KDD*, pp. 2867–2867, 2018.

Acharya, J. and Sun, Z. Communication complexity in locally private distribution estimation and heavy hitters. In *ICML*, pp. 51–60, 2019.

Ali, S. M. and Silvey, S. D. A general class of coefficients of divergence of one distribution from another. *JRS: Series B (Methodological)*, 28(1):131–142, 1966.

Apple Differential Privacy Team. Learning with privacy at scale. *Apple Machine Learning Journal*, 2017.

Balcer, V. and Cheu, A. Separating local & shuffled differential privacy via histograms. In *ITC*, pp. 1:1–1:14, 2020.

Balle, B., Bell, J., Gascón, A., and Nissim, K. The privacy blanket of the shuffle model. In *CRYPTO*, pp. 638–667, 2019.

Balle, B., Bell, J., Gascón, A., and Nissim, K. Private summation in the multi-message shuffle model. *arXiv: 2002.00817*, 2020.

Barthe, G. and Olmedo, F. Beyond differential privacy: Composition theorems and relational logic for f-divergences between probabilistic programs. In *ICALP*, pp. 49–60, 2013.

Bassily, R. and Smith, A. Local, private, efficient protocols for succinct histograms. In *STOC*, pp. 127–135, 2015.

Bassily, R., Nissim, K., Stemmer, U., and Thakurta, A. G. Practical locally private heavy hitters. In *NIPS*, pp. 2288–2296, 2017.

Beimel, A., Nissim, K., and Omri, E. Distributed private data analysis: Simultaneously solving how and what. In *CRYPTO*, pp. 451–468, 2008.

Bittau, A., Erlingsson, Ú., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnés, J., and Seefeld, B. Prochlo: Strong privacy for analytics in the crowd. In *SOSP*, pp. 441–459, 2017.

Blum, A., Dwork, C., McSherry, F., and Nissim, K. Practical privacy: the SuLQ framework. In *PODS*, pp. 128–138, 2005.

Canonne, C. A short note on Poisson tail bounds, 2017. URL http://www.cs.columbia.edu/~ccanonne/files/misc/2017-poissonconcentration.pdf.

Chan, T. H., Shi, E., and Song, D. Optimal lower bound for differentially private multi-party aggregation. In *ESA*, pp. 277–288, 2012.

Cheu, A., Smith, A. D., Ullman, J., Zeber, D., and Zhilyaev, M. Distributed differential privacy via shuffling. In *EUROCRYPT*, pp. 375–403, 2019.

Csiszár, I. Eine informationstheoretische ungleichung und ihre anwendung auf beweis der ergodizitaet von markoffschen ketten. *Magyer Tud. Akad. Mat. Kutato Int. Koezl.*, 8:85–108, 1964.

Csiszár, I. Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

Ding, B., Kulkarni, J., and Yekhanin, S. Collecting telemetry data privately. In *NIPS*, pp. 3571–3580, 2017.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pp. 486–503, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *TCC*, pp. 265–284, 2006b.

Dwork, C., Roth, A., et al. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Erlingsson, Ú., Pihur, V., and Korolova, A. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *CCS*, pp. 1054–1067, 2014.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. Amplification by shuffling: From local to central differential privacy via anonymity. In *SODA*, pp. 2468–2479, 2019.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Song, S., Talwar, K., and Thakurta, A. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *arXiv:2001.03618*, 2020.

Feller, W. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, January 1968.

Ghazi, B., Golowich, N., Kumar, R., Pagh, R., and Velingker, A. On the power of multiple anonymous messages. Cryptology ePrint Archive:1382, 2019a.

Ghazi, B., Pagh, R., and Velingker, A. Scalable and differentially private distributed aggregation in the shuffled model. *arXiv: 1906.08320*, 2019b.

Ghazi, B., Kumar, R., Manurangsi, P., Pagh, R., and Velingker, A. Pure differentially private summation from anonymous messages. In *ITC*, pp. 15:1–15:23, 2020a.

Ghazi, B., Manurangsi, P., Pagh, R., and Velingker, A. Private aggregation from fewer anonymous messages. In *EUROCRYPT*, pp. 798–827, 2020b.

Ghosh, A., Roughgarden, T., and Sundararajan, M. Universally utility-maximizing privacy mechanisms. *SICOMP*, 41(6):1673–1693, 2012.

Goryczka, S. and Xiong, L. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE TDSC*, 14(5):463–477, 2015.

Greenberg, A. Apple's "differential privacy" is about collecting your data – but not your data. *Wired, June*, 13, 2016.

Ishai, Y., Kushilevitz, E., Ostrovsky, R., and Sahai, A. Cryptography from anonymity. In *FOCS*, pp. 239–248, 2006.

Kairouz, P., Bonawitz, K., and Ramage, D. Discrete distribution estimation under local privacy. In *ICML*, pp. 2436–2444, 2016.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and open problems in federated learning. *arXiv: 1912.04977*, 2019.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Rashkodnikova, S., and Smith, A. What can we learn privately? In *FOCS*, pp. 531–540, 2008.

Kearns, M. Efficient noise-tolerant learning from statistical queries. *JACM*, 45(6):983–1006, 1998.

Knuth, D. E. *The Art of Computer Programming, Volume II: Seminumerical Algorithms, 2nd Edition*. Addison-Wesley, 1981.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv: 1610.05492*, 2016.

Kotsogiannis, I., Tao, Y., He, X., Fanaeepour, M., Machanavajjhala, A., Hay, M., and Miklau, G. PrivateSQL: a differentially private SQL query engine. *VLDB*, 12(11):1371–1384, 2019.

Kotz, S., Kozubowski, T., and Podgorski, K. *The Laplace Distribution and Generalizations: A Revisit with Applications to Communications, Economics, Engineering, and Finance.* Progress in Mathematics Series, 2001.

Ruggles, S., Flood, S., Goeken, R., Grover, J., Meyer, E., Pacas, J., and Sobek, M. Integrated public use microdata series (IPUMS) USA: Version 9.0 [dataset]. *Minneapolis, MN*, 2019. URL https://doi.org/10.18128/D010.V9.0.

Sason, I. and Verdu, S. f-divergence inequalities. *IEEE TOIT*, 62(11):5973–6006, 2016.

Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *INTERSPEECH*, pp. 1058–1062, 2014.

Shankland, S. How Google tricks itself to protect Chrome user privacy. *CNET, October*, 2014.

Stemmer, U. Locally private k-means clustering. In *SODA*, pp. 548–559, 2020.

Suresh, A. T. Differentially private anonymized histograms. In *NeurIPS*, pp. 7969–7979, 2019.

Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. Distributed mean estimation with limited communication. In *ICML*, pp. 3329–3337, 2017.

Vadhan, S. *The Complexity of Differential Privacy*, pp. 347–450. Springer International Publishing, 2017.

Wang, T., Xu, M., Ding, B., Zhou, J., Li, N., and Jha, S. MURS: Practical and robust privacy amplification with multi-party differential privacy. *arXiv:1908.11515*, 2019.

Warner, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *JASA*, 60(309):63–69, 1965.

Wilson, R. J., Zhang, C. Y., Lam, W., Desfontaines, D., Simmons-Marengo, D., and Gipson, B. Differentially private SQL with bounded user contribution. *arXiv:1909.01917*, 2019.

A. Additional Preliminaries

For a real number y, we let $y_+ := \max(y, 0)$. We next recall the hockey stick divergence which belongs to the class of f-divergences introduced by (Ali & Silvey, 1966; Csiszár, 1964; 1967).

Definition 19 (Hockey Stick Divergence; e.g., (Sason & Verdu, 2016))**.** *For any $\varepsilon > 0$, the e^ε-*hockey stick divergence *between distributions $\mathcal{D}$ and $\mathcal{D}'$ is defined as* $d_\varepsilon(\mathcal{D}\|\mathcal{D}') = \sum_{x\in\text{supp}(\mathcal{D})}[\mathcal{D}(x) - e^\varepsilon \cdot \mathcal{D}'(x)]_+$.

The following connection between hockey-stick divergence and DP was observed by (Barthe & Olmedo, 2013) and follows immediately from Definitions 5 and 19.

Lemma 20. *A mechanism $\mathcal{M}$ is (ε, δ)-DP if and only if* $\max_{\mathbf{x}\sim\mathbf{x}'} d_\varepsilon(\mathcal{M}(\mathbf{x})\|\mathcal{M}(\mathbf{x}')) \le \delta$.

B. Missing proofs from Sections 3 and 4

The following is a well-known fact that will help us determine the privacy guarantee of the $\mathcal{D}$ Mechanism for Δ-summation; it is an immediate consequence of Lemma 20.

Lemma 21. *For any distribution $\mathcal{D}$ supported on integers, the $\mathcal{D}$ Mechanism for Δ-summation is (ε, δ)-DP if and only if* $\max_{-\Delta\le k\le\Delta} d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) \le \delta$.

B.1. Proof of Theorem 11

To prove Theorem 11, we need the following concentration bound for Poisson distributions (see, e.g., (Canonne, 2017))

Lemma 22 (Poisson Concentration)**.** *For any $\lambda, y \in \mathbb{R}^+$,*

$$\Pr_{Y\sim\text{Poi}(\lambda)}[|Y - \lambda| \ge y] \le 2e^{-\frac{y^2}{2(y+\lambda)}}.$$

Proof of Theorem 11. Let $\mathcal{D} = \text{Poi}(\lambda)$ where λ is as specified in the theorem statement. From Lemma 21, it suffices to show that $d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) \le \delta$ for all $k \in \{-\Delta, \dots, \Delta\}$. To bound $d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) \le \delta$, recall that for $\mathcal{D} = \text{Poi}(\lambda)$, we have $\mathcal{D}(Y) = \frac{\lambda^Y e^{-\lambda}}{Y!}$. Hence,

$$\frac{\mathcal{D}(Y)}{\mathcal{D}(Y-k)} = \lambda^k \cdot \frac{(Y-k)!}{Y!}.$$

This implies that $\frac{\mathcal{D}(Y)}{\mathcal{D}(Y-k)} < e^{|k|\varepsilon}$ for all $Y \in [e^{-\varepsilon/|k|}\lambda + |k|, e^{\varepsilon/|k|}\lambda - |k|]$, which is a superset of $[e^{-\varepsilon/\Delta}\lambda + \Delta, e^{\varepsilon/\Delta}\lambda - \Delta]$. From this, we can bound the hockey stick divergence as follows.

$$\begin{aligned}
d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) &= \sum_{Y\in\mathbb{Z}}[\mathcal{D}(Y) - e^\varepsilon \cdot \mathcal{D}(Y-k)]_+ \\
&= \sum_{Y\in\mathbb{Z}\setminus[e^{-\varepsilon/\Delta}\lambda+\Delta, e^{\varepsilon/\Delta}\lambda-\Delta]}[\mathcal{D}(Y) - e^\varepsilon \cdot \mathcal{D}(Y-k)]_+ \\
&\le \sum_{Y\in\mathbb{Z}\setminus[e^{-\varepsilon/\Delta}\lambda+\Delta, e^{\varepsilon/\Delta}\lambda-\Delta]}\mathcal{D}(Y) \\
&= \Pr_{Y\sim\text{Poi}(\lambda)}[Y < e^{-\varepsilon/\Delta}\lambda + \Delta] + \Pr_{Y\sim\text{Poi}(\lambda)}[Y > e^{\varepsilon/\Delta}\lambda - \Delta].
\end{aligned}$$

From our choice of λ, we also have $e^{-\varepsilon/\Delta}\lambda + \Delta \le \lambda - 0.5(1-e^{-\varepsilon/\Delta})\lambda$ and $e^{\varepsilon/\Delta}\lambda - \Delta \ge \lambda + 0.5(1-e^{-\varepsilon/\Delta})\lambda$. Hence, we may apply Lemma 22 (with $y = 0.5(1 - e^{-\varepsilon/\Delta})\lambda$) which gives

$$d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) \le 2\exp\left(-\frac{0.25(1-e^{-\varepsilon/\Delta})^2\lambda^2}{4\lambda}\right) \le \delta,$$

where the last inequality follows from the fact that $\lambda \ge 16\log(2/\delta)/(1 - e^{-\varepsilon/\Delta})^2$. Hence, we conclude that the $\text{Poi}(\lambda)$ Mechanism is (ε, δ)-DP as desired. □

B.2. Proof of Theorem 13

To prove Theorem 13, we need the following generic version of the Chernoff bound; note that the bound is in fact just a Markov inequality in disguise.

Lemma 23. *For every real number $t < 0$, any real number a and any random variable X, $\Pr[X \le a] \le \mathbb{E}[e^{tX}]/e^{ta}$.*

Proof of Theorem 13. Let $\mathcal{D} = \mathrm{NB}(r, p)$ where r, p are as specified in the theorem statement. From Lemma 21, it suffices to show that $d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) \le \delta$ for all $k = \{-\Delta, \cdots, \Delta\}$. Recall that, for every $y \in \mathbb{N}$, we have

$$\begin{aligned}\mathcal{D}(y) &= \binom{y+r-1}{y} \cdot (1-p)^r p^y \\ &= \frac{(y+r-1)\cdots(r)}{y!} \cdot (1-p)^r p^y.\end{aligned}$$

To bound $d_\varepsilon(\mathcal{D}\|k + \mathcal{D})$, we consider two cases based on whether $k \ge 0$:

Case I: $k \ge 0$**.** Observe that $\mathcal{D}(y) \ge p \cdot \mathcal{D}(y-1) \ge e^{-\varepsilon/\Delta} \cdot \mathcal{D}(y-1)$ for all $y \in \mathbb{Z}$. Hence, we have $\mathcal{D}(y) \ge e^{-k\varepsilon/\Delta} \cdot \mathcal{D}(y-k) \ge e^{-\varepsilon} \cdot \mathcal{D}(y-k)$. Hence, we have $d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) = 0$.

Case II: $k < 0$**.** In this case, observe that for $y \ge \frac{r}{e^{0.9\varepsilon/\Delta}-1}$, we have

$$\begin{aligned}\frac{\mathcal{D}(y)}{\mathcal{D}(y-k)} &= \frac{(y-k+r-1)\cdots(y+r)}{(y-k+1)\cdots(y+1)} \cdot p^k \\ &\le \left(\frac{y+r}{y+1} \cdot \frac{1}{p}\right)^{-k} \\ \left(\because y \ge \frac{r}{e^{0.9\varepsilon/\Delta}-1}\right) &\le (e^{\varepsilon/\Delta})^{-k} \\ &\le e^{\varepsilon}.\end{aligned}$$

As a result, we have

$$\begin{aligned}d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) &= \sum_{y\in\mathbb{Z}} [\mathcal{D}(y) - e^\varepsilon \cdot \mathcal{D}(y-k)]_+ \\ &= \sum_{y\in\mathbb{Z}, y < \frac{r}{e^{0.9\varepsilon/\Delta}-1}} [\mathcal{D}(y) - e^\varepsilon \cdot \mathcal{D}(y-k)]_+ \\ &= \Pr_{Y\sim\mathcal{D}}\left[Y < \frac{r}{e^{0.9\varepsilon/\Delta}-1}\right].\end{aligned}$$

It is well-known that for $t < -\ln p$, we have $\mathbb{E}_{X\sim\mathcal{D}}[e^{tX}] = \left(\frac{1-p}{1-pe^t}\right)^r$. By applying Lemma 23 with $a = \frac{r}{e^{0.9\varepsilon/\Delta}-1}$ and $t = -0.8\varepsilon/\Delta$, we have

$$\begin{aligned}d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) &\le \frac{\left(\frac{1-p}{1-pe^t}\right)^r}{e^{t\left(\frac{r}{e^{0.9\varepsilon/\Delta}-1}\right)}} \\ &= \left(\frac{1-p}{1-pe^t} \cdot \exp\left(\frac{-t}{e^{0.9\varepsilon/\Delta}-1}\right)\right)^r \\ &= \left(\frac{1-e^{-0.1\varepsilon/\Delta}}{1-e^{-0.9\varepsilon/\Delta}} \cdot \exp\left(\frac{0.8\varepsilon/\Delta}{e^{0.9\varepsilon/\Delta}-1}\right)\right)^r, \qquad (1)\end{aligned}$$

where we used $t = -0.8\varepsilon/\Delta$ and $p = e^{-0.1\varepsilon/\Delta}$. We will now consider two further subcases, based on ε/Δ.

- Case IIa: $\varepsilon/\Delta \le 1$. Recall the identity $e^x \ge 1 + x$ for all $x \in \mathbb{R}$. From this, we have

  $$\frac{0.8\varepsilon/\Delta}{e^{0.9\varepsilon/\Delta}-1} \le \frac{0.8\varepsilon/\Delta}{0.9\varepsilon/\Delta} \le 1,$$

  and

  $$1 - e^{-0.1\varepsilon/\Delta} \le 0.1\varepsilon/\Delta,$$

  and

  $$\begin{aligned}1 - e^{-0.9\varepsilon/\Delta} &= 1 - \frac{1}{e^{0.9\varepsilon/\Delta}} \ge 1 - \frac{1}{1+0.9\varepsilon/\Delta} \\ &= \frac{0.9\varepsilon/\Delta}{1+0.9\varepsilon/\Delta} \ge 0.4\varepsilon/\Delta,\end{aligned}$$

  where the last inequality follows from our assumption that $\varepsilon/\Delta \le 1$.

  Plugging the above three inequalities back into (1),

  $$d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) \le \left(\frac{1}{4} \cdot e\right)^r \le \delta,$$

  where the inequality follows from our choice of $r \ge 50\log(1/\delta)$.

- Case IIb: $\varepsilon/\Delta > 1$. In this case, we may use the same inequality as before to derive

  $$\begin{aligned}\frac{1-e^{-0.1\varepsilon/\Delta}}{1-e^{-0.9\varepsilon/\Delta}} &= 1 - \frac{e^{-0.1\varepsilon/\Delta} - e^{-0.9\varepsilon/\Delta}}{1-e^{-0.9\varepsilon/\Delta}} \\ &= 1 - \frac{e^{0.8\varepsilon/\Delta}-1}{e^{0.9\varepsilon/\Delta}-1} \\ (\because 1 + x \le e^x) &\le \exp\left(-\frac{e^{0.8\varepsilon/\Delta}-1}{e^{0.9\varepsilon/\Delta}-1}\right).\end{aligned}$$

  Plugging this back into (1), we arrive at

  $$d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) \le \exp\left(-r \cdot \frac{e^{0.8\varepsilon/\Delta}-1-0.8\varepsilon/\Delta}{e^{0.9\varepsilon/\Delta}-1}\right).$$

  Now, recall that the Taylor expansion of e^x is $\sum_{k=0}^\infty \frac{x^k}{k!}$. Hence, for $x \ge 0$, we have $e^x \ge 1 + x + 0.5x^2$. Using this to the term $e^{0.8\varepsilon/\Delta}$, we get

  $$\begin{aligned}d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) &\le \exp\left(-r \cdot \frac{0.32(\varepsilon/\Delta)^2}{e^{0.9\varepsilon/\Delta}-1}\right) \\ (\because \varepsilon/\Delta > 1) &\le \exp\left(-r \cdot 0.32e^{-\varepsilon/\Delta}\right) \\ (\because r \ge 50\log(1/\delta) \cdot e^{\varepsilon/\Delta}) &\le \delta.\end{aligned}$$

Hence, in both cases, we get $d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) \le \delta$ as desired. □

B.3. Proof of Lemma 15

To prove Lemma 15, we will resort to Feller's characterization (Feller, 1968) of infinitely divisible distribution as a discrete compound Poisson distribution. We begin by recalling the definition of the latter.

Definition 24. *A distribution $\mathcal{D}$ is said to be a* discrete compound Poisson (DCP) *distribution if there exists a distribution $\mathcal{D}'$ on positive integers and a non-negative real number λ such that the following process results in the random variable Y being distributed as $\mathcal{D}$. First, generate $N \sim \mathrm{Poi}(\lambda)$. Then, let $X_1, \ldots, X_N$ be i.i.d. random variables distributed as $\mathcal{D}'$. Finally, let $Y = X_1 + \cdots + X_N$.*

When this condition holds, we write $\mathcal{D} = \mathrm{DCP}(\lambda, \mathcal{D}')$.

A fundamental theorem, due to Feller (Feller, 1968), states that every infinitely divisible distribution $\mathcal{D}$ on non-negative integers is a DCP distribution:

Theorem 25 ((Feller, 1968))**.** *Every infinitely divisible distribution $\mathcal{D}$ on non-negative integers is a discrete compound Poisson distribution.*

The above theorem implies the following observation:

Observation 26. *For any infinitely divisible distribution $\mathcal{D}$ on non-negative integers,* $\mathrm{Var}(\mathcal{D}) \geq \mathbb{E}[\mathcal{D}]$.

Proof. From Theorem 25, $\mathcal{D} = \mathrm{DCP}(\lambda, \mathcal{D}')$ for some $\lambda \geq 0$ and a distribution $\mathcal{D}'$ on positive integers. It is well-known that $\mathbb{E}[\mathcal{D}] = \lambda \cdot \mathbb{E}_{X\sim\mathcal{D}'}[X]$ and $\mathrm{Var}[\mathcal{D}] = \lambda \cdot \mathbb{E}_{X\sim\mathcal{D}'}[X^2]$. Since $\mathcal{D}'$ is on positive integers, we must have $\mathbb{E}_{X\sim\mathcal{D}'}[X] \leq \mathbb{E}_{X\sim\mathcal{D}'}[X^2]$, which implies that $\mathbb{E}[\mathcal{D}] \leq \mathrm{Var}[\mathcal{D}]$. □

We are now ready to prove Lemma 15.

Proof of Lemma 15. From Lemma 21, we have $d_\varepsilon(\mathcal{D} \| 1 + \mathcal{D}) \leq \delta$, which can be rearranged as

$$\delta \geq \sum_{i=0}^{\infty} [\mathcal{D}(i) - e^\varepsilon \cdot \mathcal{D}(i-1)]_+,$$

which implies that $\mathcal{D}(i) \leq e^\varepsilon \cdot \mathcal{D}(i-1) + \delta$ for all non-negative integer i. From this and from $\mathcal{D}$ is supported from non-negative integer (i.e. $\mathcal{D}(-1) = 0$), it is simple to show via induction that $\mathcal{D}(i) \leq \delta \cdot \left(\frac{e^{(i+1)\varepsilon}-1}{e^\varepsilon - 1}\right)$. Let $j = \lfloor \frac{1}{\varepsilon}\left(\ln\left(\frac{1}{\delta}\right) - \ln\left(\frac{2}{\varepsilon^2}\right)\right) \rfloor - 1$; we have

$$\begin{aligned}
\Pr_{Y\sim\mathcal{D}}[Y < j] &\leq \frac{\delta}{e^\varepsilon - 1} \cdot \left(\sum_{i=0}^{j-1} (e^{(i+1)\varepsilon} - 1)\right) \\
&\leq \frac{\delta}{e^\varepsilon - 1} \cdot \left(\frac{e^{(j+1)\varepsilon}}{e^\varepsilon - 1}\right) \\
&\leq \frac{\delta}{\varepsilon^2} \cdot e^{(j+1)\varepsilon} \\
&\leq \frac{1}{2},
\end{aligned}$$

where the last inequality follows from our choice of j. Thus, we have $\mathbb{E}[\mathcal{D}] \geq j \cdot \frac{1}{2} \geq \Omega_\varepsilon(\log(1/\delta))$. Invoking Observations 26 and 9 immediately yields the desired result. □

B.4. Proof of Theorem 18

Proof of Theorem 18. Observe that the analyzer's view only contains (U_{+1}, U_{-1}). Since there is a one-to-one correspondence between (U_{+1}, U_{-1}) and $(U_{+1} - U_{-1}, U_{-1})$, we may consider the analyzer's view as the latter instead; for convenience, let $U_{\text{diff}} = U_{+1} - U_{-1}$. Note that the distribution of $(U_{\text{diff}}, U_{-1})$ (of course) depends on the input data set $\mathbf{x} = (x_1, \ldots, x_i)$; when we would like to stress this, we write $U^{\mathbf{x}}_{\text{diff}}, U^{\mathbf{x}}_{-1}$ instead of the usual U_{diff}, U_{-1}.

Let Z_i^1, Z_i^2, Z_i^3 be the random variables that the user samples. Define $\hat{Z}^1, \hat{Z}^2, \hat{Z}^3$ as $\sum_{i\in[n]} Z_i^1, \sum_{i\in[n]} Z_i^2, \sum_{i\in[n]} Z_i^3$ respectively. We have

$$U_{+1} = \left(\sum_{i\in[n]} x_i\right) + \hat{Z}^1 + \hat{Z}^3,$$

and

$$U_{-1} = \hat{Z}^2 + \hat{Z}^3. \tag{2}$$

Hence, we also have

$$U_{\text{diff}} = \left(\sum_{i\in[n]} x_i\right) + \hat{Z}^1 - \hat{Z}^2. \tag{3}$$

Moreover, from how Z_i^1, Z_i^2, Z_i^3's are sampled, we have that $\hat{Z}^1, \hat{Z}^2, \hat{Z}^3$ are independent random variables with distributions $\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3$ respectively.

Next, consider $(x_1, \ldots, x_n) = \mathbf{x} \sim \mathbf{x}' = (x'_1, \ldots, x'_n)$. We can calculate the $e^{\varepsilon_1+\varepsilon_2}$-hockey stick divergence between $(U^{\mathbf{x}}_{\text{diff}}, U^{\mathbf{x}}_{-1}), (U^{\mathbf{x}'}_{\text{diff}}, U^{\mathbf{x}'}_{-1})$ as follows:

$$\begin{aligned}
&d_{\varepsilon_1+\varepsilon_2}((U^{\mathbf{x}}_{\text{diff}}, U^{\mathbf{x}}_{-1}) \| (U^{\mathbf{x}'}_{\text{diff}}, U^{\mathbf{x}'}_{-1})) \qquad (4)\\
&= \sum_{a,b\in\mathbb{Z}} \Big[\Pr[U^{\mathbf{x}}_{\text{diff}} = a, U^{\mathbf{x}}_{-1} = b] \\
&\qquad - e^{\varepsilon_1+\varepsilon_2} \cdot \Pr[U^{\mathbf{x}'}_{\text{diff}} = a, U^{\mathbf{x}'}_{-1} = b] \Big]_+ \\
&= \sum_{a,b\in\mathbb{Z}} \Big[\Pr[U^{\mathbf{x}}_{\text{diff}} = a] \Pr[U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a] \\
&\qquad - e^{\varepsilon_1+\varepsilon_2} \cdot \Pr[U^{\mathbf{x}'}_{\text{diff}} = a] \Pr[U^{\mathbf{x}'}_{-1} = b \mid U^{\mathbf{x}'}_{\text{diff}} = a] \Big]_+ \\
&= \sum_{a,b\in\mathbb{Z}} \Pr[U^{\mathbf{x}'}_{\text{diff}} = a] \cdot \Big[\frac{\Pr[U^{\mathbf{x}}_{\text{diff}} = a]}{\Pr[U^{\mathbf{x}'}_{\text{diff}} = a]} \cdot \Pr[U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a]
\end{aligned}$$

$$- e^{\varepsilon_1+\varepsilon_2} \cdot \Pr[U^{\mathbf{x}'}_{-1} = b \mid U^{\mathbf{x}'}_{\text{diff}} = a]\Big]_+$$

Now, observe that $\Pr[U^{\mathbf{x}}_{\text{diff}} = a]$ and $\Pr[U^{\mathbf{x}'}_{\text{diff}} = a]$ are the probability that the $(\mathcal{D}_1 - \mathcal{D}_2)$ mechanism outputs a on input $\mathbf{x}$ and $\mathbf{x}'$ respectively. Since we assume that the $(\mathcal{D}_1 - \mathcal{D}_2)$ mechanism is ε_1-DP, the ratio between the two must not exceed e^{ε_1}. Plugging this into the above equation, we have

$$\begin{aligned}
&d_{\varepsilon_1+\varepsilon_2}((U^{\mathbf{x}}_{\text{diff}}, U^{\mathbf{x}}_{-1}) \| (U^{\mathbf{x}'}_{\text{diff}}, U^{\mathbf{x}'}_{-1})) \\
&\le \sum_{a,b\in\mathbb{Z}} e^{\varepsilon_1} \cdot \Pr[U^{\mathbf{x}'}_{\text{diff}} = a] \cdot \Big[\Pr[U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a] \\
&\qquad - e^{\varepsilon_2} \cdot \Pr[U^{\mathbf{x}'}_{-1} = b \mid U^{\mathbf{x}'}_{\text{diff}} = a]\Big]_+ \\
&= \sum_{a\in\mathbb{Z}} e^{\varepsilon_1} \cdot \Pr[U^{\mathbf{x}'}_{\text{diff}} = a] \cdot \sum_{b\in\mathbb{Z}} \Big[\Pr[U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a] \\
&\qquad - e^{\varepsilon_2} \Pr[U^{\mathbf{x}'}_{-1} = b \mid U^{\mathbf{x}'}_{\text{diff}} = a]\Big]_+ \qquad (5)
\end{aligned}$$

We now rearrange the term $\Pr[U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a]$ as:

$$\begin{aligned}
&\Pr[U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a] \\
&= \sum_{c\in\mathbb{Z}_{\ge 0}} \Pr[\hat{Z}^2 = c \wedge U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a] \\
&= \sum_{c\in\mathbb{Z}_{\ge 0}} \Pr[\hat{Z}^2 = c \wedge \hat{Z}^3 = b - c \mid U^{\mathbf{x}}_{\text{diff}} = a] \\
&= \sum_{c\in\mathbb{Z}_{\ge 0}} \Pr[\hat{Z}^2 = c \wedge \hat{Z}^3 = b - c \mid U^{\mathbf{x}}_{\text{diff}} = a] \\
&= \sum_{c\in\mathbb{Z}_{\ge 0}} \Pr[\hat{Z}^3 = b - c] \cdot \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a] \\
&= \sum_{c\in\mathbb{Z}_{\ge 0}} \mathcal{D}^3(b - c) \cdot \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a],
\end{aligned}$$

where the third line follows since $\hat{Z}^3$ is independent of $(\hat{Z}^2, U_{\text{diff}})$. Similarly, we have

$$\begin{aligned}
&\Pr[U^{\mathbf{x}'}_{-1} = b \mid U^{\mathbf{x}'}_{\text{diff}} = a] \\
&= \sum_{c'\in\mathbb{Z}_{\ge 0}} \mathcal{D}^3(b - c') \cdot \Pr[\hat{Z}^2 = c' \mid U^{\mathbf{x}'}_{\text{diff}} = a].
\end{aligned}$$

Notice also that $\sum_{c\in\mathbb{Z}_{\ge 0}} \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a] = \sum_{c'\in\mathbb{Z}_{\ge 0}} \Pr[\hat{Z}^2 = c' \mid U^{\mathbf{x}'}_{\text{diff}} = a] = 1$. Plugging this into the two previous equations, we have

$$\begin{aligned}
\Pr[U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a] = &\sum_{c,c'\in\mathbb{Z}_{\ge 0}} \mathcal{D}^3(b - c) \\
&\cdot \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a] \Pr[\hat{Z}^2 = c' \mid U^{\mathbf{x}'}_{\text{diff}} = a],
\end{aligned}$$

and

$$\begin{aligned}
\Pr[U^{\mathbf{x}'}_{-1} = b \mid U^{\mathbf{x}'}_{\text{diff}} = a] = &\sum_{c,c'\in\mathbb{Z}_{\ge 0}} \mathcal{D}^3(b - c') \\
&\cdot \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a] \Pr[\hat{Z}^2 = c' \mid U^{\mathbf{x}'}_{\text{diff}} = a].
\end{aligned}$$

These imply that

$$\begin{aligned}
&\sum_{b\in\mathbb{Z}} \Big[\Pr[U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a] - e^{\varepsilon_2} \cdot \Pr[U^{\mathbf{x}'}_{-1} = b \mid U^{\mathbf{x}'}_{\text{diff}} = a]\Big]_+ \\
&= \sum_{b\in\mathbb{Z}} \Big[\sum_{c,c'\in\mathbb{Z}_{\ge 0}} \big(\mathcal{D}^3(b - c) - e^{\varepsilon_2} \cdot \mathcal{D}^3(b - c')\big) \\
&\qquad \cdot \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a] \Pr[\hat{Z}^2 = c' \mid U^{\mathbf{x}'}_{\text{diff}} = a]\Big]_+ \\
&\le \sum_{b\in\mathbb{Z}} \sum_{c,c'\in\mathbb{Z}_{\ge 0}} \Big[\big(\mathcal{D}^3(b - c) - e^{\varepsilon_2} \cdot \mathcal{D}^3(b - c')\big) \\
&\qquad \cdot \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a] \Pr[\hat{Z}^2 = c' \mid U^{\mathbf{x}'}_{\text{diff}} = a]\Big]_+ \\
&= \sum_{c,c'\in\mathbb{Z}_{\ge 0}} \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a] \Pr[\hat{Z}^2 = c' \mid U^{\mathbf{x}'}_{\text{diff}} = a] \\
&\qquad \cdot \sum_{b\in\mathbb{Z}} \big[\mathcal{D}^3(b - c) - e^{\varepsilon_2} \cdot \mathcal{D}^3(b - c')\big]_+ \\
&= \sum_{c,c'\in\mathbb{Z}_{\ge 0}} \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a] \Pr[\hat{Z}^2 = c' \mid U^{\mathbf{x}'}_{\text{diff}} = a] \\
&\qquad \cdot \sum_{b\in\mathbb{Z}} \big[\mathcal{D}^3(b) - e^{\varepsilon_2} \cdot \mathcal{D}^3(b + (c - c'))\big]_+ \\
&= \sum_{c,c'\in\mathbb{Z}_{\ge 0}} \Pr[\hat{Z}^2 = c \mid U^{\mathbf{x}}_{\text{diff}} = a] \Pr[\hat{Z}^2 = c' \mid U^{\mathbf{x}'}_{\text{diff}} = a] \\
&\qquad \cdot d_{\varepsilon_2}(\mathcal{D}^3 \| (c - c') + \mathcal{D}^3).
\end{aligned}$$

Recall our assumption that the $\mathcal{D}^3$ mechanism for Δ-summation is $(\varepsilon_2, \delta_2)$-DP. From Lemma 21, we have $d_{\varepsilon_2}(\mathcal{D}^3 \| (c - c') + \mathcal{D}^3) \le \delta_2$ for all $c, c' \in \{0, \ldots, \Delta\}$. Thus, we can further bound the above expression as

$$\begin{aligned}
&\sum_{b\in\mathbb{Z}} \Big[\Pr[U^{\mathbf{x}}_{-1} = b \mid U^{\mathbf{x}}_{\text{diff}} = a] - e^{\varepsilon_2} \cdot \Pr[U^{\mathbf{x}'}_{-1} = b \mid U^{\mathbf{x}'}_{\text{diff}} = a]\Big]_+ \\
&\le \delta_2 + \Pr[\hat{Z}^2 > \Delta \mid U^{\mathbf{x}}_{\text{diff}} = a] + \Pr[\hat{Z}^2 > \Delta \mid U^{\mathbf{x}'}_{\text{diff}} = a].
\end{aligned}$$

Plugging the above into (5) yields

$$\begin{aligned}
&d_{\varepsilon_1+\varepsilon_2}((U^{\mathbf{x}}_{\text{diff}}, U^{\mathbf{x}}_{-1}) \| (U^{\mathbf{x}'}_{\text{diff}}, U^{\mathbf{x}'}_{-1})) \\
&\le \sum_{a\in\mathbb{Z}} e^{\varepsilon_1} \cdot \Pr[U^{\mathbf{x}'}_{\text{diff}} = a] \cdot \Big(\delta_2 + \Pr[\hat{Z}^2 > \Delta \mid U^{\mathbf{x}}_{\text{diff}} = a] \\
&\qquad + \Pr[\hat{Z}^2 > \Delta \mid U^{\mathbf{x}'}_{\text{diff}} = a]\Big) \\
&\le e^{\varepsilon_1} \cdot \delta_2 + e^{2\varepsilon_1} \cdot \Pr[\hat{Z}^2 > \Delta] + e^{\varepsilon_1} \cdot \Pr[\hat{Z}^2 > \Delta] \\
&\le e^{\varepsilon_1} \cdot \delta_2 + 2e^{2\varepsilon_1} \cdot \delta_3,
\end{aligned}$$

where the second inequality follows from $\Pr[U^{\mathbf{x}'}_{\text{diff}} = a] \le e^{\varepsilon_1} \cdot \Pr[U^{\mathbf{x}}_{\text{diff}} = a]$ which in turn holds due to the ε_1-DP of the $(\mathcal{D}_1 - \mathcal{D}_2)$ Mechanism, and the last inequality follows directly from the concentration assumption on the noise $\mathcal{D}_2$. As a result, from Lemma 20, the $(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$-Correlated Distributed Mechanism is $(\varepsilon_1 + \varepsilon_2, e^{\varepsilon_1} \cdot \delta_2 + 2e^{2\varepsilon_1} \cdot \delta_3)$-DP in the shuffled model as desired. □

B.5. Proof of Theorem 1

Proof of Theorem 1. Let us pick our parameters as follows:

- $\varepsilon_1 = (1-\gamma)\varepsilon$ and $\varepsilon_2 = \gamma\varepsilon$.
- $\delta_2 = \delta_3 = \frac{\delta}{e^{\varepsilon_1}+2e^{2\varepsilon_1}}$
- $\Delta = \lceil \frac{\log(1/\delta_2)}{\varepsilon_1} \rceil$.
- Let r, p be as in Theorem 13 with $\varepsilon = \varepsilon_2, \delta = \delta_2$, i.e.,

$$p = e^{-0.1\varepsilon_2/\Delta} \text{ and } r = 50 \cdot e^{\varepsilon_2/\Delta} \cdot \log\left(\frac{1}{\delta_2}\right).$$

We simply run the protocol $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Mechanism for $\mathcal{D}^1 = \mathcal{D}^2 = \mathrm{NB}(1, e^{-\varepsilon_1})$ and $\mathcal{D}^3 = \mathrm{NB}(r, p)$. We now argue that this protocol yields the desired privacy, accuracy, and (expected) communication.

Privacy. We will apply Theorem 18. To do so, we have to show that the three required conditions are satisfied:

- (Privacy of True Noise) Since $\mathcal{D}^1 - \mathcal{D}^2 = \mathrm{DLap}(\varepsilon_1)$, the $(\mathcal{D}^1 - \mathcal{D}^2)$ Mechanism is ε_1-DP in the Central model.
- (Privacy of Correlated Noise) From our choice of r, p, Theorem 13 immediately implies that the $\mathcal{D}^3$ Mechanism is $(\varepsilon_2, \delta_2)$-DP for Δ-summation in the Central model.
- (Concentration of Negative Noise) We may compute $\Pr_{Y\sim\mathcal{D}^2}[Y > \Delta]$ as follows:

$$\Pr_{Y\sim\mathcal{D}^3}[Y > \Delta] = \sum_{i=\Delta+1}^{\infty} \mathcal{D}^2(i) = \sum_{i=\Delta+1}^{\infty} (1-e^{-\varepsilon_1})e^{-\varepsilon_1 i}$$
$$= e^{-\varepsilon_1(\Delta+1)} \le \delta_3,$$

where the last inequality follows from our choice of Δ.

Hence, we can apply Theorem 18 which implies that the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Mechanism is $(\varepsilon_1 + \varepsilon_2, e^{\varepsilon_1}\cdot\delta_2 + 2e^{2\varepsilon_1}\cdot\delta_3)$-DP in the shuffled model. Since $\varepsilon_1 + \varepsilon_2 = \varepsilon$ and $e^{\varepsilon_1}\cdot\delta_2 + 2e^{2\varepsilon_1}\cdot\delta_3 = \delta$, this is indeed the desired privacy guarantee.

Accuracy. From Observation 16, the MSE of the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Mechanism is $\mathrm{Var}(\mathcal{D}^1) + \mathrm{Var}(\mathcal{D}^2) = \mathrm{Var}(\mathcal{D}^1 - \mathcal{D}^2) = \mathrm{Var}(\mathrm{DLap}((1-\gamma)\varepsilon_1))$ as desired.

Communication. From Observation 17, the expected number of messages sent by each user is at most

$$1 + \frac{\mathbb{E}[\mathcal{D}^1] + \mathbb{E}[\mathcal{D}^2] + \mathbb{E}[\mathcal{D}^3]}{n} \le 1 + \frac{e^{\varepsilon_1} + e^{\varepsilon_1} + r/(1-p)}{n}$$
$$\le 1 + O\left(\Delta \cdot \frac{\log(1/\delta)}{\varepsilon_2 n}\right)$$
$$\le 1 + O\left(\frac{\log(1/\delta)^2}{\varepsilon_1\varepsilon_2 n}\right)$$
$$\le 1 + O\left(\frac{\log(1/\delta)^2}{\gamma\varepsilon^2 n}\right),$$

where the second inequality follows from the choice of p, r and since $\varepsilon \le O(1)$, the third inequality follows from the choice of Δ, and the last inequality follows from the choice of $\varepsilon_1, \varepsilon_2$ and since $\gamma \le 1/2$. □

C. From Binary Summation to Histograms

In this section, we prove Corollary 2 which we start by recalling:

Corollary 27 (Histograms). *For every $\varepsilon \le O(1)$ and every $\delta, \gamma \in (0, 1/2)$, there is an (ε, δ)-DP protocol for histograms on sets of size B in the multi-message shuffled model, with error equal to a vector of independent discrete Laplace random variables each with parameter $\frac{(1-\gamma)\varepsilon}{2}$ and with an expected communication per user of $(\log B + 1)\cdot(1 + O(\frac{B\log^2(1/\delta)}{\gamma\varepsilon^2 n}))$ bits.*

Algorithm 5 Histogram Randomizer.

procedure HISTOGRAMRANDOMIZER$_{\mathcal{R}}(i)$
  For $j = 1$ **to** B:
    $S_j \leftarrow \mathcal{R}(\mathbf{1}[i = j])$
    $R_j \leftarrow \{j\} \times S_j$
  return $\bigcup_{j=1}^{B} R_j$

Algorithm 6 Histogram Analyzer.

procedure HISTOGRAMANALYZER$_{\mathcal{A}}(R)$
  For $j = 1$ **to** B:
    $R_j \leftarrow \{y_1 \mid y \in R \text{ and } y_0 = j\}$
    $a_j \leftarrow \mathcal{A}(R_j)$
  return $(a_j)_{j=1}^{B}$

Proof of Corollary 27. We define the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Histogram Randomizer by instantiating the generic histogram analyzer given in Algorithm 5 with $\mathcal{R} = \text{RANDOMIZER}_{\mathcal{D}^1,\mathcal{D}^2,\mathcal{D}^3,n}$ from Algorithm 3. Similarly, we define $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Histogram Analyzer by instantiating the generic histogram analyzer given in Algorithm 6 with $\mathcal{A} = \text{ANALYZER}_{\mathcal{D}^1,\mathcal{D}^2}$ from Algorithm 4. Thus, the resulting procedures proceed via the same templates as the binary summation randomizer and analyzer in Algorithms 3 and 4 respectively but by applying it to each of the B buckets. As a consequence, each message consists of an index in $[B]$ in addition to the increment/decrement bit which yields a

communication cost per message of $\log B + 1$ bits. We next provide the privacy, accuracy, and communication analyses for completeness. To do so, we start by defining the ∞-div distributions $\mathcal{D}^1$, $\mathcal{D}^2$, and $\mathcal{D}^3$ to be used in the aforementioned calls to Algorithms 5 and 6. Given ε, δ and γ as in Corollary 27, we define the parameters:

- $\varepsilon_1 = \frac{(1-\gamma)\varepsilon}{2}$ and $\varepsilon_2 = \frac{\gamma\varepsilon}{2}$.
- $\delta_2 = \delta_3 = \frac{\delta}{2\cdot(e^{\varepsilon_1}+2e^{2\varepsilon_1})}$
- $\Delta = \lceil \frac{\log(1/\delta_2)}{\varepsilon_1} \rceil$.
- Let r, p be as in Theorem 13 with $\varepsilon = \varepsilon_2, \delta = \delta_2$, i.e.,

$$p = e^{-0.1\varepsilon_2/\Delta} \text{ and } r = 50 \cdot e^{\varepsilon_2/\Delta} \cdot \log\left(\frac{1}{\delta_2}\right).$$

We set $\mathcal{D}^1 = \mathcal{D}^2 = \mathrm{NB}(1, e^{-\varepsilon_1})$ and $\mathcal{D}^3 = \mathrm{NB}(r, p)$. Note that these are the same settings as in the proof of Theorem 1 except that ε is replaced by $\frac{\varepsilon}{2}$ and δ is replaced by $\frac{\delta}{2}$.

Privacy. Note that in the shuffled model, the analyzer in Algorithm 6 only observes the number of $+1$'s and the number of -1's for each of the B buckets. We will first prove the privacy guarantee in the case where the analyzer only observes the number of $+1$'s and the number of -1's for a *single bucket*, and then extend the argument to the case where it observes all B buckets. For a fixed bucket, the task reduces to the privacy of the bit summation protocol which was shown in the proof of Theorem 1. Due to our slightly different setting of parameters where ε and δ are replaced by $\frac{\varepsilon}{2}$ and $\frac{\delta}{2}$ respectively, this implies that the protocol is $(\frac{\varepsilon}{2}, \frac{\delta}{2})$-DP. Extending to the case of multiple buckets, we note that any change in a single user's input would affect exactly *two* buckets. Using the fact that the noise variables in these two buckets are independent, the Basic Composition theorem (see, e.g., Theorem B.1 of (Dwork et al., 2014)), this implies that in the general case where the analyzer observes the counts for all buckets, the histograms protocol given by Algorithms 5 and 6 is (ε, δ)-DP as desired.

Accuracy. Note that the B-dimensional error vector in Algorithm 6 consists of independent coordinates each equal to the unbiased difference of two independent $\mathrm{NB}(1, e^{-\varepsilon_1})$ random variables. Thus, each coordinate of the error vector is distributed according to $\mathrm{DLap}(\varepsilon_1) = \mathrm{DLap}((1-\gamma)\varepsilon/2)$ as desired.

Communication. The expected number of messages sent by each user in Algorithm 5 is

$$\begin{aligned} &1 + B \cdot (\mathbb{E}[\mathcal{D}^1_{/n}] + \mathbb{E}[\mathcal{D}^2_{/n}] + 2 \cdot \mathbb{E}[\mathcal{D}^3_{/n}]) \\ &= 1 + B \cdot \left(\frac{\mathbb{E}[\mathcal{D}^1] + \mathbb{E}[\mathcal{D}^2] + 2 \cdot \mathbb{E}[\mathcal{D}^3]}{n}\right) \end{aligned} \tag{6}$$

But $\mathbb{E}[\mathcal{D}^1] = \mathbb{E}[\mathcal{D}^2] = \frac{e^{-\varepsilon_1}}{1-e^{-\varepsilon_1}} = \frac{1}{e^{\varepsilon_1}-1} \le \frac{1}{\varepsilon_1}$. On the other hand,

$$\begin{aligned} \mathbb{E}[\mathcal{D}^3] &= \frac{pr}{1-p} = \frac{e^{-0.1\varepsilon_2/\Delta} \cdot \log(1/\delta_2) \cdot 50 \cdot e^{\varepsilon_2/\Delta}}{1 - e^{-0.1\varepsilon_2/\Delta}} \\ &= \frac{50 \cdot e^{\varepsilon_2/\Delta} \cdot \log(1/\delta_2)}{e^{0.1\varepsilon_2/\Delta} - 1} \le \frac{500 \cdot \Delta \cdot e^{\varepsilon_2/\Delta} \cdot \log(1/\delta_2)}{\varepsilon_2}. \end{aligned}$$

Plugging back in Equation (6) and using the facts that $\Delta = \lceil \frac{\log(1/\delta_2)}{\varepsilon_1} \rceil$, $\varepsilon_2 = \frac{\gamma\varepsilon}{2}$, $\delta_2 = \frac{\delta}{2\cdot(e^{\varepsilon_1}+2e^{2\varepsilon_1})}$, and $\varepsilon = O(1)$, we get that the expected number of messages sent by each user is at most

$$1 + B \cdot \left(\frac{2}{\varepsilon_1 n} + O\left(\frac{\log^2(1/\delta)}{\varepsilon_1 \varepsilon_2 n}\right)\right) = 1 + O\left(\frac{B \log^2(1/\delta)}{\gamma \varepsilon^2 n}\right).$$

The expected number of bits of communication per user stated in Corollary 27 now follows using the fact that each message in Algorithm 5 consists of $\log B + 1$ bits. □

Improving the Computational Efficiency of Randomizer. If implemented naively, the randomizer (Algorithm 5) has a running time that depends linearly on B, since it has to go over all the buckets $i \in [B]$ and run the randomizer on each bucket, which corresponds to sampling random variables Z_i^1, Z_i^2, Z_i^3. Intuitively, this should not be necessary since most of these random variables are zero. Below, we will show that this intuition is indeed correct, in the sense that the randomizer can be significantly sped up, leading to an expected running time that is linear in the expected per-user communication cost.[5]

To do so, let us abstract the problem at hand as follows:

Definition 28. *Let $\mathcal{D}$ be an infinitely divisible distribution on non-negative integers. In the* ParallelSampling *problem for $\mathcal{D}$, we are given two positive integers n, B and the goal is to output a multiset S whose elements are from $[B]$ such that, if we let Y_i be the number of occurrences of i in S, then $Y_1, \ldots, Y_B$ are distributed as i.i.d. $\mathcal{D}_{/n}$ random variables.*

Recall from Theorem 25 that every infinitely divisible distribution on non-negative integers can be represented as a discrete compound Poisson (DCP) distribution. We will exploit this representation to give an efficient algorithm for ParallelSampling, as stated below.

Theorem 29. *Let $\mathcal{D}$ be the discrete compound Poisson distribution $\mathrm{DCP}(\lambda, \mathcal{D}')$. Suppose that there is an algorithm $\mathcal{A}'$ that can sample $\mathcal{D}'$ in expected time T. Then, $\text{ParallelSampling}_{\mathcal{D}}(n, B)$ can be performed in expected time $O\left(1 + \mathbb{E}[\mathcal{D}] \cdot \frac{B \log B}{n} + \lambda \cdot T \cdot \frac{B}{n}\right)$.*

[5] For ease of exposition, we assume a model of computation where the logarithm and exponential functions can be computed in constant time, and where a uniform random variable in $(0, 1)$ can be sampled in constant time.

Algorithm 7 Efficient Sampling for $\mathcal{D} = \mathrm{DCP}(\lambda, \mathcal{D}')$.

1: **procedure** PARALLELSAMPLING$_{\mathcal{D}}(n, B)$
2: $S \leftarrow \{\}$
3: Sample $N_{\text{sum}} \sim \mathrm{Poi}(B\lambda/n)$
4: **For** $\ell = 1$ **to** N_{sum}:
5: Randomly sample $j \sim [B]$
6: Use algorithm $\mathcal{A}'$ to sample $X \sim \mathcal{D}'$
7: Add X copies of j to S
8: **return** S

We will need the following fact about generating Poisson random variables:

Lemma 30 ((Knuth, 1981))**.** *There is an algorithm that takes a positive real number λ and generates a sample from $\mathrm{Poi}(\lambda)$ in expected $O(1+\lambda)$ steps.*

We are now ready to prove Theorem 29.

Proof of Theorem 29. Our sampling procedure is given in Algorithm 7.

Correctness. For every $i \in [B]$, let N_i denote the number of iterations for which the index j sampled in line 7 is equal to i. From standard properties[6] of the Poisson random variable, we have $N_1, \ldots, N_B$ are i.i.d. $\mathrm{Poi}(\lambda/n)$ random variables. As a result, the algorithm is equivalent to: for every $i \in [B]$, independently generate $N_i \sim \mathrm{Poi}(\lambda/n)$, generate $X_1, \ldots, X_{N_i} \sim \mathcal{D}'$ and let $Y_i = X_1 + \cdots + X_{N_i}$. In other words, we have $Y_1, \ldots, Y_B$ are i.i.d. $\mathrm{DCP}(\lambda/n, \mathcal{D}') = \mathrm{DCP}(\lambda, \mathcal{D}')_{/n} = \mathcal{D}_{/n}$ as desired.

Expected Running Time. We calculate the expected running time for each step of our algorithm:

- Line 7 clearly takes constant time.
- For line 7, we can use Lemma 30 to sample from $\mathrm{Poi}(B\lambda/n)$ in expected time $O\left(1 + \frac{B\lambda}{n}\right)$.
- Notice that $\mathbb{E}[N_{\text{sum}}] = B\lambda/n$. Thus, each of the following sub-steps is repeated this many times in expectation. We now list the running time for each sub-step:
  - Line 7 takes at most $O(\log B)$ time.
  - From our assumption on $\mathcal{A}'$, line 7 takes T time.
  - Line 7 takes $O(\mathbb{E}[\mathcal{D}'] \cdot \log B)$ time in expectation.

[6] Specifically, it is well-known that if we first pick $N_{\text{sum}} \sim \mathrm{Poi}(\tilde{\lambda})$ and let $N_1, \ldots, N_n$ be a multinomial distribution with N_{sum} trials and the probability of incrementing i-th bucket in each trial is p_i, then N_i's are independent $\mathrm{Poi}(\tilde{\lambda} \cdot p_i)$ random variables. This is sometimes referred to in literature as *Poissonization*.

Hence, in total, the expected running time of Algorithm 7 is:

$$
\begin{aligned}
&O\left(1 + \frac{B\lambda}{n} \cdot (\log B + T + \mathbb{E}[\mathcal{D}'] \cdot \log B)\right) \\
&= O\left(1 + \frac{B\lambda T}{n} + \lambda \cdot \mathbb{E}[\mathcal{D}'] \cdot \frac{B \log B}{n}\right) \\
&= O\left(1 + \frac{B\lambda T}{n} + \mathbb{E}[\mathcal{D}] \cdot \frac{B \log B}{n}\right),
\end{aligned}
$$

where the first equality used that $\mathbb{E}[\mathcal{D}'] = \Omega(1)$ which follows from the fact that $\mathcal{D}'$ is supported on the positive integers. This completes our proof. □

Using our $\mathcal{D}$-PARALLELSAMPLING routine and its analysis in Theorem 29, we are now ready to describe our computationally efficient implementation of the histogram randomizer. The pseudo-code is given in Algorithm 8.

Algorithm 8 Efficient Histogram Randomizer.

1: **procedure** HISTOGRAMRANDOMIZER$_{\mathcal{D}^1,\mathcal{D}^2,\mathcal{D}^3,n,B}(i)$
2: $S_1 \leftarrow$ PARALLELSAMPLING$_{\mathcal{D}^1}(n, B)$
3: $S_2 \leftarrow$ PARALLELSAMPLING$_{\mathcal{D}^2}(n, B)$
4: $S_3 \leftarrow$ PARALLELSAMPLING$_{\mathcal{D}^3}(n, B)$
5: $R \leftarrow \{(i, +1)\}$
6: **For** $j \in S_1$:
7: Add $(j, +1)$ to R
8: **For** $j \in S_2$:
9: Add $(j, -1)$ to R
10: **For** $j \in S_3$:
11: Add both $(j, +1)$ and $(j, -1)$ to R
12: **return** R

Note that the running time of this algorithm is on the same order as the time need to run PARALLELSAMPLING on distributions $\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3$ and inputs n, B. Since $\mathcal{D}^1 = \mathcal{D}^2 = \mathrm{NB}(1, e^{-\varepsilon_1})$ and $\mathcal{D}^3 = \mathrm{NB}(r, p)$ in our proof of Corollary 27, our task boils down to using an efficient sampling procedure for the Negative Binomial distribution. Since $\mathrm{NB}(r, p)$ is infinitely divisible, Theorem 25 implies that it is a discrete compound Poisson distribution. In this case, the corresponding distribution $\mathcal{D}'$ on positive integers from Definition 24 is known to be the *logarithmic distribution* with parameter p. We will apply Theorem 29 with $\mathcal{D} = \mathrm{NB}(r, p)$ and $\mathcal{D}'$ set to the logarithmic distribution with parameter p. To do so, we need to upper-bound the time needed to sample from the logarithmic distribution. This is achieved in the following lemma.

Lemma 31. *There is an algorithm that takes a parameter $p \in (0, 1)$ and generates a sample from the logarithmic distribution with parameter p in an expected number of steps proportional to the mean of the distribution.*

Proof. Recall that the probability mass function of the logarithmic distribution with parameter $p \in (0,1)$ is given by $f(k) := -\frac{1}{\ln(1-p)}\frac{p^k}{k}$ for any positive integer k. We will apply inverse transform sampling. Namely, let $F(\cdot)$ denote the cumulative distribution function. We proceed as follows:

1. Sample a uniform number u between 0 and 1.
2. For $k = 2, 3, \ldots,$
   (a) If $F(k) > u$, return k.

Note that this procedure outputs k whenever u lands in the interval $[F(k-1), F(k))$. The probability of this event is equal the width of the interval, which is equal to the probability $f(k)$. Moreover, the expected number of iterations run in the for loop is equal to the mean of the distribution. Note as we iterate over increasing values of k in the above procedure, the value of $F(k)$ can be computed from that of $F(k-1)$ in constant time. This follows from the identity $f(k) = f(k-1) \cdot p \cdot \frac{(k-1)}{k}$ which holds fro all positive integers k larger than 1. □

By Theorem 29 and Lemma 31, we conclude that using the implementation in Algorithm 8 allows us to reduce the expected running time of the histogram randomizer from $\Omega(B)$ to $O\left(1 + \frac{B\log^2(1/\delta)}{\gamma\varepsilon^2 n}\right)$ (i.e, to the same order as the expected per user communication complexity).

D. Lower Bound for Single-Message Binary Summation

We now prove a lower bound against single-message protocols for binary summation (Theorem 3). The lower bound does not depend on any restriction on the size of the message output by the randomizer. In fact, we prove a slightly stronger statement than in Theorem 3, in that our lower bound holds even for $\varepsilon = \ln n - \log n - \Theta(1)$, as stated more precisely below.

Theorem 32. *Let $\delta = 1/n^c$ and $\varepsilon \le \ln n - \ln(d \log n)$ for any positive constants c and d. Then, any (ε, δ)-DP protocol for Binary Summation in the single-message shuffled model must incur an expected squared error of at least $f \log n$ for some positive constant f depending on c and d.*

To prove Theorem 32, we will use the following result.

Lemma 33 ((Balle et al., 2019), Lemma 4.1)**.** *Let $\mathcal{M} = (\mathcal{R}, \mathcal{A})$ be an n-party protocol for Binary Summation in the single-message shuffled model. Assume that the inputs $X_1, \ldots, X_n$ to the n parties are sampled i.i.d. from some distribution on $\{0,1\}$. Then, there is a protocol $\mathcal{M}' = (\mathcal{R}', \mathcal{A}')$ in the single-message shuffled model s.t.:*

1. *Each message from the randomizer to the shuffler is a real number in the interval $[0,1]$ and the analyzer simply sums its n incoming messages. Namely, $\mathrm{Im}(\mathcal{R}') \subseteq [0,1]$ and $\mathcal{A}'(y_1, \ldots, y_n) = \sum_{i=1}^n y_i$.*
2. *The MSE of the protocol $\mathcal{M}'$ is at most that of the protocol $\mathcal{M}$ (with respect to the i.i.d. input distribution).*
3. *If the protocol $\mathcal{M}$ is (ε, δ)-DP, then the protocol $\mathcal{M}'$ is (ε, δ)-DP.*

We are now ready to prove Theorem 32.

Proof of Theorem 32. Let $\mathcal{M} = (\mathcal{R}, \mathcal{A})$ be an (ε, δ)-DP protocol for Binary Summation in the single-message shuffled protocol with expected MSE α. Consider the case where the inputs of the n users are drawn i.i.d. from the uniform distribution on $\{0,1\}$. By Lemma 33, there is an (ε, δ)-DP protocol $\mathcal{M}' = (\mathcal{R}', \mathcal{A}')$ in the single-message shuffled model where the randomizer outputs a real number in $[0,1]$, the analyzer sums all n incoming messages, and where the MSE of $\mathcal{M}'$ is at most that of $\mathcal{M}$. For each $b \in \{0,1\}$, let $\mathcal{D}_b$ be the probability distribution supported over a subset T_b of the real line and corresponding to the output of the randomizer $\mathcal{R}'$ on input b. Let $T := T_0 \cup T_1$. We next lower bound the total variation distance between $\mathcal{D}_0$ and $\mathcal{D}_1$ in terms of α. Specifically, we show that $d_{TV}(\mathcal{D}_0, \mathcal{D}_1) \ge 1 - \frac{8\alpha}{n}$. Since[7] $d_{TV}(\mathcal{D}_0, \mathcal{D}_1) := 1 - \sum_{x \in T} \min\{\mathcal{D}_0(x), \mathcal{D}_1(x)\}$, this is equivalent to proving that

$$\sum_{x \in T} \min\{\mathcal{D}_0(x), \mathcal{D}_1(x)\} \le \frac{8\alpha}{n}. \tag{7}$$

To prove this inequality, note that the MSE of the protocol $\mathcal{M}'$ satisfies

$$\begin{aligned}
\alpha^2 &= \mathbb{E}\left[\left(\sum_{i\in[n]}(x_i - y_i)\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i\in[n]}\sum_{j\in[n]}(x_i - y_i)(x_j - y_j)\right] \\
&= \sum_{i\in\in[n]}\mathbb{E}[(x_i - y_i)^2] + \sum_{i\neq j\in[n]}\mathbb{E}[(x_i - y_i)(x_j - y_j)] \\
&\overset{(*)}{=} n \cdot \mathbb{E}[(x-y)^2] + (n^2 - n)\cdot\mathbb{E}[x-y]^2 \\
&\ge n \cdot \mathbb{E}[(x-y)^2] \\
&= \frac{n}{2}\cdot(\mathbb{E}[y^2 \mid x = 0] + \mathbb{E}[(1-y)^2 \mid x = 1]) \\
&= \frac{n}{2}\cdot\left(\sum_{y\in T_0}\mathcal{D}_0(y)y^2 + \sum_{y\in T_1}\mathcal{D}_1(y)(1-y)^2\right) \\
&= \frac{n}{2}\cdot\sum_{y\in T}(\mathcal{D}_0(y)y^2 + \mathcal{D}_1(y)(1-y)^2)
\end{aligned}$$

[7] We assume that the messages sent are encoded as bit strings, meaning that T is countable and $\sum_{x\in T}\min\{\mathcal{D}_0(x), \mathcal{D}_1(x)\}$ is well-defined.

$$\geq \frac{n}{8} \cdot \sum_{y \in T} \min\{\mathcal{D}_0(y), \mathcal{D}_1(y)\},$$

where the equality (*) uses the i.i.d. property of the user inputs and the local randomizers. Inequality (7) now follows.

Let $\delta = 1/n^c$ for any positive constant c, and let ε be any value satisfying $\varepsilon \leq \ln n - \ln(d \cdot \log n)$ for some positive constant d. We next show that the protocol $\mathcal{M}'$ violates (ε, δ)-DP whenever $\alpha < f \cdot \log n$, where f is a positive constant that depends on c and d. To do so, we define S_0 to be the set of all values of $y \in T_0$ for which $\mathcal{D}_0(y) > \mathcal{D}_1(y)$ and similarly define S_1 as the set of all values of $y \in T_1$ for which $\mathcal{D}_1(y) > \mathcal{D}_0(y)$. By definition, S_0 and S_1 are disjoint. Let $\kappa := (8\alpha)/n$. We next argue that $\mathcal{D}_0(S_0) \geq 1 - \kappa$. By symmetry, it would follow that $\mathcal{D}_1(S_1) \geq 1 - \kappa$. Assume for the sake of contradiction that $\mathcal{D}_0(S_0) < 1 - \kappa$. Then, $\mathcal{D}_0(T \setminus S_0) > \kappa$. In this case, we have that

$$\begin{aligned}
\sum_{y \in T} \min\{\mathcal{D}_0(y), \mathcal{D}_1(y)\} &\geq \sum_{y \in (T \setminus S_0)} \min\{\mathcal{D}_0(y), \mathcal{D}_1(y)\} \\
&\geq \sum_{y \in (T \setminus S_0)} \mathcal{D}_0(y) \\
&= \mathcal{D}_0(T \setminus S_0) \\
&> \kappa \\
&= \frac{8\alpha}{n},
\end{aligned}$$

which would contradict (7).

To show that the protocol $\mathcal{M}'$ violates (ε, δ)-DP, we consider two neighboring length-n binary input sequences: $\mathbf{x} = (0, 0, \ldots, 0)$ and $\mathbf{x}' = (1, 0, \ldots, 0)$. To show that $\mathcal{M}'$ violates (ε, δ)-DP, it suffices to show that:

$$\Pr[\mathcal{R}(\mathbf{x}) \subseteq S_0] > e^{\varepsilon} \cdot \Pr[\mathcal{R}(\mathbf{x}') \subseteq S_0] + \delta.$$

We first note that:

$$\begin{aligned}
\Pr[\mathcal{R}(\mathbf{x}) \subseteq S_0] &= \mathcal{D}_0(S_0)^n \\
&\geq (1 - \kappa)^n \\
&= \left(1 - \frac{8\alpha}{n}\right)^n \\
&\geq e^{-8\alpha - O(\frac{\alpha^2}{n})}.
\end{aligned}$$

On the other hand, we have that:

$$\begin{aligned}
\Pr[\mathcal{R}(\mathbf{x}') \subseteq S_0] &= \mathcal{D}_0(S_0)^{n-1} \cdot \mathcal{D}_1(S_0) \\
&\leq \mathcal{D}_0(S_0)^{n-1} \cdot \mathcal{D}_1(T \setminus S_1) \\
&\leq \mathcal{D}_0(S_0)^{n-1} \cdot \kappa.
\end{aligned}$$

Thus,

$$\begin{aligned}
\frac{\Pr[\mathcal{R}(\mathbf{x}) \subseteq S_0]}{\Pr[\mathcal{R}(\mathbf{x}') \subseteq S_0]} &\geq \frac{\mathcal{D}_0(S_0)^n}{\mathcal{D}_0(S_0)^{n-1}\kappa} \\
&= \frac{\mathcal{D}_0(S_0)}{\kappa} \\
&\geq \frac{1 - \kappa}{\kappa} \\
&= \frac{n}{8\alpha} - 1.
\end{aligned}$$

So it suffices to choose α such that $e^{-8 \cdot \alpha - O(\frac{\alpha^2}{n})} > 2\delta$ and $\frac{n}{8\alpha} - 1 > 2e^{\varepsilon}$. For $\delta = 1/n^c$, where c is a positive constant, and for ε satisfying $\varepsilon \leq \ln n - \ln(d \log n)$, where d is a positive constant, we can satisfy the last two inequalities and get a violation of (ε, δ)-DP as long as $\alpha < f \log n$ where f is a positive constant depending on c and d. □

E. Experiments for Histogram

We next evaluate our algorithms for histogram. Once again, we consider the Correlated Distributed Histogram Mechanism[8] and the Poisson Histogram Mechanism[9].

Apart from our algorithms, we also consider three algorithms, each of which can be considered a generalization of the Randomized Response mechanism in the binary case. Each of the three algorithms is parameterized by a "noise parameter" p. Below we summarize how they work.

- **B-Randomized Response (B-RR).** In B-RR, the randomizer outputs the input bin x with probability $1 - p$. With the remaining probability p, it simply outputs a random bin from $1, \ldots, B$.
- **RAPPOR.** In RAPPOR (Erlingsson et al., 2014), the randomizer first encodes the input x using its one-hot encoding s, which is simply a B-bit string whose xth coordinate is the only one with value 1. Then, the randomizer flips each bit of s independently at random with probability p, and output the resulting string $\tilde{s}$.
- **Fragmented RAPPOR.** While both B-RR and RAPPOR are single-message, Fragmented RAPPOR (Erlingsson et al., 2020)[10] can be thought of as the multiple-message version of RAPPOR. The randomizer works similarly to RAPPOR, except that, instead of outputting the final string $\tilde{s}$ as a single message, it outputs B messages $(1, \tilde{s}_1), \ldots, (B, \tilde{s}_B)$.

[8] This is simply Algorithms 5, 6 with $\mathcal{R}, \mathcal{A}$ being the randomizers and analyzers of the Correlated Distributed Mechanism.

[9] This is simply Algorithms 5, 6 with $\mathcal{R}, \mathcal{A}$ being the randomizers and analyzers of the Poisson Distributed Mechanism.

[10] We remark what we are using here is what (Erlingsson et al., 2020) called "Attributed-fragmented RAPPOR". They also introduce another fragmentation technique (called "report fragmentation"). To the best of our knowledge, however, this other fragmentation technique only helps in terms of privacy in their model when there are multiple shufflers, and hence is irrelevant to our setting.

While the naive imeplementation of Fragmented RAPPOR results in the randomizer always outputting B messages, (Erlingsson et al., 2020) noted that we can instead sends only the coordinates of $\tilde{s}$ that are 1. In this case, the expected number of messages sent per user becomes $(1-p)+p(B-1) = 1+p(B-2)$. In other words, the expected number of messages overhead is $p(B-2)$.

A slightly different algorithm was independently proposed in (Ghazi et al., 2019a). In our experiments, the algorithm of (Ghazi et al., 2019a) and Fragmented RAPPOR produce essentially the same results (less than 1% difference in errors and numbers of messages overhead) and hence we do not specifically discuss the former further here.

In all three algorithms, the analyzer just output the unbiased estimator for the count of each bucket.

We remark that, to the best of our knowledge, there is no known efficient algorithm that can accurately compute parameters for the three algorithms above. Due to this, we use "optimistic" estimates for the parameters, which means that the errors/messages overhead seen in plots before for them could be better than the true numbers. We stress that, while we use "optimistic" estimates for these three algorithms, we use very accurate, "pessimistic" estimates for our own algorithms. The detailed explanation on how these parameters are computed is given in Appendix G.

Unlike our algorithms, the ℓ_∞ error of Randomized Respose, RAPPOR and Fragmented RAPPOR are data-dependent. For the experiments, we use two datasets from IPUMS which are available online (Ruggles et al., 2019). The remainder of this section is divided based on the two datasets, which will be explained in more details below.

For each selection of parameters $\varepsilon, \delta, n, B$ and a dataset specified below, we run 100 repetitions of each algorithm. We then record the RMSE and the average ℓ_∞ errors over these repetitions. In all experiments we run below, the errors (both RMSE and ℓ_∞) of RAPPOR is significantly larger than other algorithms; specifically, RAPPOR's errors are always larger than $2.5\times$ that of RR which is the second-worst algorithm in terms of errors. As a result, we do not include RAPPOR in the plots for readability.

We also remark that, in all the plots below, it will be the case that Fragmented RAPPOR and Poisson are essentially the same, both in terms of errors and expected number of messages overhead, for any reasonably small values of δ. (This is because of the same reason as the binary case where Poisson and binary RR roughly coincide.) Due to this, we will only refer to the Poisson Mechanism in the discussions below; it should be understood that any statement that applies to the Poisson Mechanism also essentially applies to Fragmented RAPPOR. For large values of δ, it turns out Poisson Mechanism is better than Fragmented RAPPOR, both in terms of errors and number of messages overhead (see Figures 4 and 5).

E.1. IPUMS City Dataset.

The first dataset we consider for histogram computation is the city distribution for the US population in the 1940 census. This dataset was used for evaluating private histogram algorithms in the shuffled model by (Wang et al., 2019). As in the previous work, we discard data points with an unidentified city. Doing so, we get $n = 60,313,201$ reports and $B = 915$ cities.

Similar to our experiments in binary summation, we study the effect of varying ε, δ on the errors and the number of messages overhead per user.

Errors. The plots for ℓ_∞ errors and RMSEs as we vary ε and δ are included in Figure 4. The general trends of the errors for Central, Correlated Distributed and Poisson are very similar to that of the RMSEs in our binary summation experiments (Figure 2), with the B-RR algorithm always incurring noticably larger errors. One interesting difference between the histogram case here and the binary case is that the gaps between Central/Correlated Distributed and Poisson are smaller for histograms. In fact, when δ is large (e.g. $\delta \geq 10^{-4}$ for $\varepsilon = 1$), the Poisson Mechanism even incurs slightly lower ℓ_∞ errors compared to the Correlated Distributed Mechanism. (See the top-left plot in Figure 2.) However, once δ becomes sufficiently small, the expected behaviour is observed (i.e. Poisson incurs noticeably more error compared to Correlated Distributed).

Communication Complexity. We next present our plots for the expected number of additional messages sent for each user in Figure 5. The general trends are exactly the same as those shown in Figure 3 for binary summation. We also note here that, even in the rather extreme case where $\varepsilon = 0.1$ and $\delta = 2 \cdot 10^{-9}$, the expected number of messages for Correlated Distributed is only 0.181 and for Poisson is only 0.074, both of which are quite small. For moderate value of $\varepsilon = 1$, the numbers dropped to 0.021 and 0.010 respectively.

E.2. IPUMS House Value Dataset.

Since in the previous dataset the buckets correspond to a categorical feature (namely, the city), it cannot be naturally used to assess the performance of algorithms in terms of a varying number B of buckets. To do so, we also consider the house value distribution of the US population in the 1940 census. After discarding the data points with unavailable house values, we get $n = 14,958,304$ reports with

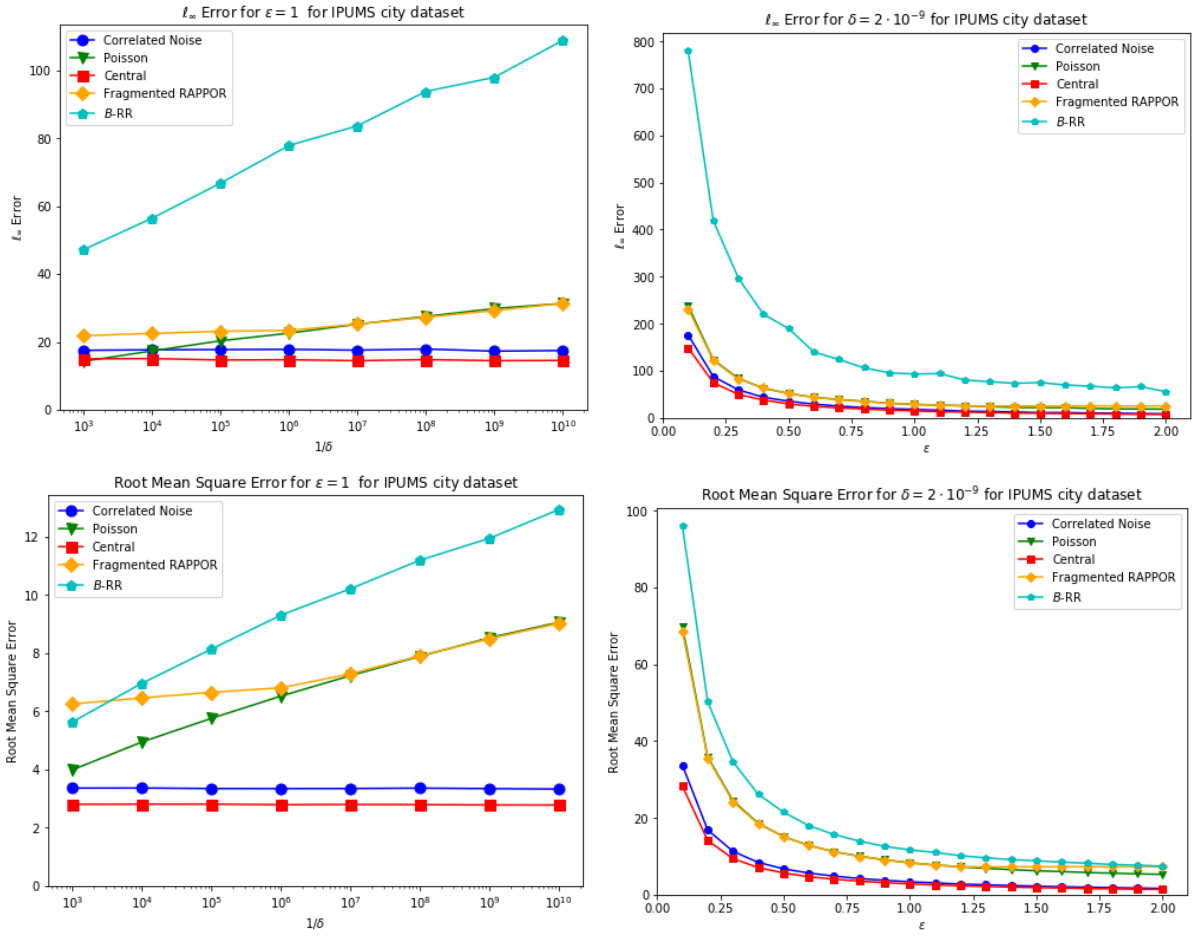

Figure 4. RMSEs and ℓ_∞ errors of different mechanisms on IPUMS city dataset.

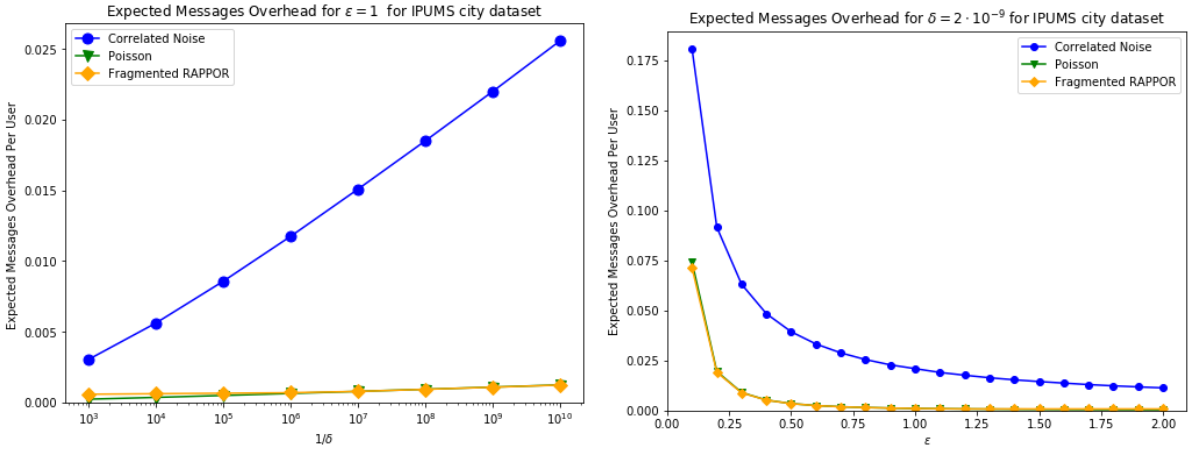

Figure 5. Expected additional number of messages sent by each user for IPUMS city dataset

house values between 1 and $9,750,975$. We then divide the range $[0, 100, 000)$ into buckets of equal width and use the width as a knob to vary the number B of buckets in our experiments. More specifically, for a given B, we divide the interval $[0, 100, 000)$ equally into $B - 1$ buckets, and we put all reports with values at least $100, 000$ into the last bucket. (There are only 6009 such reports.)

Errors. The errors as B varies from 200 to 5000 are shown in Figure 6. For ℓ_∞ error, theory predicts that the error for RR increases as $\Omega(B^{1/4})$ in the regime of parameters we consider (see e.g. (Ghazi et al., 2019a)) whereas the errors of Central, Correlated Mechanism, and Poisson only increased by $O(\log B)$. This is clearly reflected in the left plot in Figure 6. On the other hand, the RMSE remains essentially constants as even B varies; see the right plot in Figure 6.

Communication Complexity. The number of communication overhead scales linearly in terms of B, as shown in Figure 7.

F. Parameters Computation for Binary Summation Protocols

Poisson Mechanism. It is simple to see that the $\mathrm{Poi}(\lambda)$ mechanism becomes more private as λ increases[11]. We use binary search to find the smallest λ such that the $\mathrm{Poi}(\lambda)$ mechanism is (ε, δ)-DP. This check can be done efficiently by checking the condition in Lemma 21. Note that, by selecting the smallest possible λ, we are minimizing both the MSE and the expected number of messages sent, since $\mathbb{E}[\mathrm{Poi}(\lambda)] = \mathrm{Var}(\mathrm{Poi}(\lambda)) = \lambda$.

Correlated Distributed Noise Mechanism. We pick $\mathcal{D}^1 = \mathcal{D}^2 = \mathrm{NB}(1, e^{-\varepsilon_1})$ similar to the proof of Theorem 1. Here we choose ε_1 such that the root MSE (RMSE) of the protocol is 20% more than that of the (central) $\mathrm{DLap}(\varepsilon)$ Mechanism. Once we pick this ε_1 (and hence $\mathcal{D}^1, \mathcal{D}^2$), we attempt to find r, p such that, when we set $\mathcal{D}^3 = \mathrm{NB}(r, p)$, the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Mechanism is (ε, δ)-DP and that it minimizes the expected number of messages sent (or equivalently minimizes $\mathbb{E}[\mathrm{NB}(r, p)] = \frac{rp}{1-p}$). Now, for a specific r, finding the smallest $p = p^*(r)$ such that the Correlated Distributed Mechanism is (ε, δ)-DP is simple and, similar to Poisson, can be done via a binary search over p; note that here we have to compute the expression (4), instead of the expression in Lemma 21) used for Poisson Mechanism.

On the other hand, we do not know how to efficiently compute $\min_{r \in \mathbb{R}^+} \frac{rp^*(r)}{1-p^*(r)}$. Hence, we resort to generic optimizers from the `scipy` package to attempt to find this. Since we are not guaranteed to find the optimum, it is possible that the optimum number of messages can be even smaller than shown below.

Randomized Response. Similar to Poisson Mechanism, we use binary search on p; for a fixed p, we can efficiently check whether the protocol is (ε, δ)-DP using Lemma 20.

[11] This just means that if the mechanism was (ε, δ)-DP, it remains so after λ is increased.

G. Parameters Computation for Histogram Protocols

In this section, we detail how the parameters are calculated for the histogram experiments in Appendix E.

G.1. Correlated Distributed Histogram Mechanism

From the proof of Corollary 27, it suffices to select $\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3$ so that the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Mechanism is $(\varepsilon/2, \delta/2)$-DP. Similar to the binary summation case, we start by picking $\mathcal{D}^1 = \mathcal{D}^2 = \mathrm{NB}(1, e^{-\varepsilon_1})$ where ε_1 is selected so that the RMSE of the protocol is 20% more than that of $\mathrm{DLap}(\varepsilon/2)$. Once again, we use a similar approach as before to attempt to find r, p such that, when setting $\mathcal{D}^3 = \mathrm{NB}(r, p)$, the $(\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3)$-Correlated Distributed Mechanism is $(\varepsilon/2, \delta/2)$-DP and the expected number of additional messages send is as small as possible.

G.2. Poisson Histogram Mechanism

To accurately compute the parameter λ needed for the $\mathrm{Poi}(\lambda)$ Histogram Mechanism to be (ε, δ)-DP. We first prove the following exact characterization of this condition; note that here we use $\mathcal{D} \otimes \mathcal{D}'$ to denote the product distribution between $\mathcal{D}$ and $\mathcal{D}'$.

Lemma 34. *For any $\varepsilon, \delta, \lambda \geq 0$, the $\mathrm{Poi}(\lambda)$ Histogram Mechanism is (ε, δ)-DP in the shuffled model if and only if*

$$d_\varepsilon(\mathrm{Poi}(\lambda) \otimes \mathrm{Poi}(\lambda) \| (1 + \mathrm{Poi}(\lambda)) \otimes (-1 + \mathrm{Poi}(\lambda))) \leq \delta \tag{8}$$

Proof. Let $f_{hist} : [B]^n \to (\mathbb{N} \cup \{0\})^B$ denote the function that computes a histogram, i.e., $f_{hist}(x_1, \ldots, x_n) = (\sum_{j \in [n]} \mathbf{1}[x_j = i])_{i \in [B]}$. In a similar vein as Observation 8, it is simple to see that the view (after shuffling) of the $\mathrm{Poi}(\lambda)$ Histogram Mechanism is exactly the same as if we apply the (B-dimensional) central $\mathrm{Poi}(\lambda)$ Mechanism to f_{hist}. For brevity, let $\mathcal{M}$ denote the latter mechanism.

To calculate the DP parameters for the mechanism, consider two neighboring data sets $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{x}' = (x_1, \ldots, x_{n-1}, x'_n)$. Due to symmetry, we may assume that $x_n = 1$ and $x'_n = 2$. For notational convenience, let $\mathbf{a}$ denote the $f_{hist}(\mathbf{x})$, i.e. the histogram constructed from input $\mathbf{x}$. We have

$$\begin{aligned}
&d_\varepsilon(\mathcal{M}(\mathbf{x}) \| \mathcal{M}(\mathbf{x}')) \\
&= \sum_{\mathbf{y} \in \mathbb{Z}^B} [\Pr[\mathcal{M}(\mathbf{x}) = \mathbf{a} + \mathbf{y}] - e^\varepsilon \Pr[\mathcal{M}(\mathbf{x}') = \mathbf{a} + \mathbf{y}]]_+ \\
&= \sum_{\mathbf{y} \in \mathbb{Z}^B} [\Pr[Y_1 = y_1, \ldots, Y_B = y_B] - \\
&\quad e^\varepsilon \Pr[Y_1 = y_1 + 1, Y_2 = y_2 - 1, Y_3 = y_3, \ldots, Y_B = y_B]]_+.
\end{aligned}$$

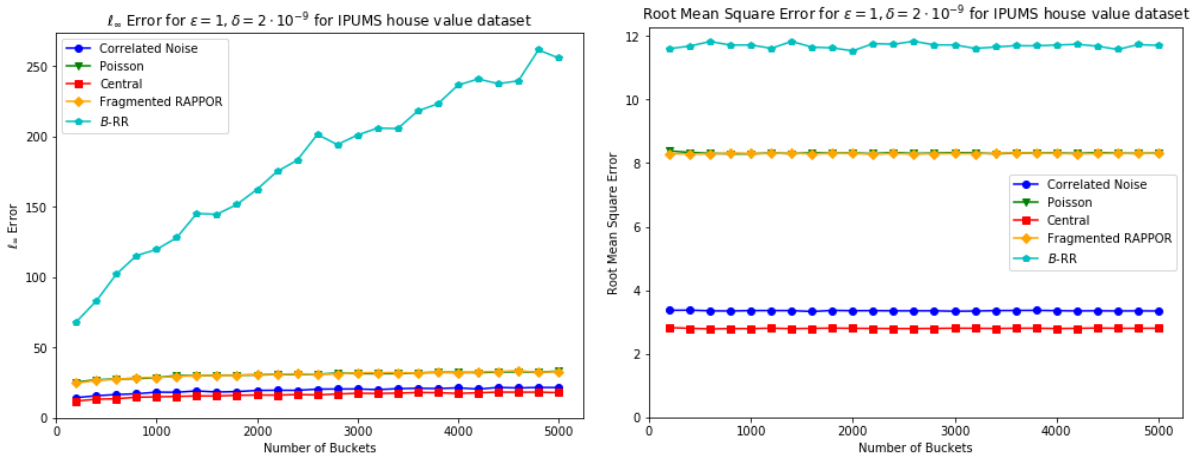

Figure 6. RMSEs and ℓ_∞ errors of different mechanisms on IPUMS house value dataset.

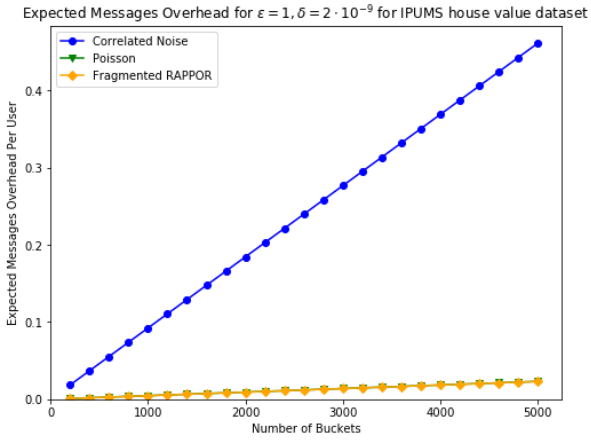

Figure 7. Expected additional number of messages sent by each user for IPUMS house value dataset

Now, since $Y_1, \ldots, Y_B$ are independent $\mathrm{Poi}(\lambda)$ random variables, we can further rewrite the above as

$$\begin{aligned}
&\sum_{\mathbf{y} \in \mathbb{Z}^B} \Pr[Y_3 = y_3, \ldots, Y_B = y_B] \cdot [\Pr[Y_1 = y_1, Y_2 = y_2] \\
&\qquad - e^\varepsilon \Pr[Y_1 = y_1 + 1, Y_2 = y_2 - 1]]_+ \\
&= \sum_{y_1, y_2 \in \mathbb{Z}} [\Pr[Y_1 = y_1, Y_2 = y_2] \\
&\qquad - e^\varepsilon \Pr[Y_1 = y_1 + 1, Y_2 = y_2 - 1]]_+ \\
&= d_\varepsilon \left(\mathrm{Poi}(\lambda) \otimes \mathrm{Poi}(\lambda) \| (1 + \mathrm{Poi}(\lambda)) \otimes (-1 + \mathrm{Poi}(\lambda)) \right).
\end{aligned}$$

From Lemma 20, we have completed our proof. □

With the above lemma in mind, we simply binary search on λ with the check being condition (8). Once again, we remark that the left hand side of (8) can be computed to arbitrary accuracy quite efficiently.

G.3. Fragmented RAPPOR

Unlike the parameters for our own protocols, we will use "optimistic" parameters for the remaining protocols, which means that the errors and expected numbers of messages overhead shown in our plots might be smaller than the actual values. For Fragmented RAPPOR, it is not hard to see that the algorithm becomes more private as p increases for any $p \in [0, 1/2]$. Hence, we may use binary search to find the noise parameter p. To do so, we only need a subroutine that, for a given p, decides whether the mechanism is (ε, δ)-DP in the shuffled model. Our algorithm will be "optimistic". In the sense that, if the algorithm says NO, then the mechanism is not (ε, δ)-DP. However, if our algorithm says YES, then the mechanism may still *not* be (ε, δ)-DP.

The checking task above further reduce to the following: given p, ε, find a lower bound on δ. Specifically, we can prove a lemma similar to Lemma 34; the main difference is that the implication is one-way instead of two-way ("if and

only if") as in Lemma 34. However, this weaker implication is already sufficient to give a lower bound on δ.

Lemma 35. *For any $p \in (0,1)$ and any positive integer n, let $\mathcal{D}_{n,p}$ denote the distribution $\mathrm{Ber}(1-p) + \mathrm{Bin}(n-1,p)$, where $\mathrm{Ber}(1-p)$ denote the Bernoulli distribution with success probability $1-p$. For any $\varepsilon, \delta > 0$, if Fragmented RAPPOR is (ε,δ)-DP in the shuffled model for n users and $B \geq 3$ buckets, then*

$$d_\varepsilon(\mathcal{D}_{n,p} \otimes \mathrm{Bin}(n,p) \| \mathrm{Bin}(n,p) \otimes \mathcal{D}_{n,p}) \leq \delta. \quad (9)$$

The proof is very similar to Lemma 34 with $\mathbf{x} = (B,\dots,B,1)$ and $\mathbf{x}' = (B,\cdots,B,2)$, and when we restrict to only the first two coordinates. We do not repeat the full argument here. We only note that the difference in quantifier comes because here we choose $\mathbf{x},\mathbf{x}'$ ourselves, whereas the previous proof of Lemma 34 works for any neighboring $\mathbf{x},\mathbf{x}'$.

We note that the left hand side in (9) can be computed in $O(n^2)$ by writing it as

$$d_\varepsilon(\mathcal{D}_{n,p} \otimes \mathrm{Bin}(n,p) \| \mathrm{Bin}(n,p) \otimes \mathcal{D}_{n,p}) \quad (10)$$

$$= \sum_{i=0}^{n}\sum_{j=0}^{n} [\mathcal{D}_{n,p}(i)\,\mathrm{Bin}(j;n,p) - e^\varepsilon \cdot \mathrm{Bin}(i;n,p)\mathcal{D}_{n,p}(j)]_+, \quad (11)$$

where we use $\mathrm{Bin}(i;n,p)$ to denote the probability mass of $\mathrm{Bin}(n,p)$ at i. We can now compute $d_\varepsilon(\mathcal{D}_{n,p} \otimes \mathrm{Bin}(n,p) \| \mathrm{Bin}(n,p) \otimes \mathcal{D}_{n,p})$ in $O(n^2)$ time, by enumerating $i,j \in \{0,\dots,n\}$ and compute the inner term. (Note that we can precompute factorials so that computing each $\mathrm{Bin}(\cdot;n,p), \mathcal{D}_{n,p}(\cdot)$ takes only $O(1)$ time.)

In our settings of parameters (e.g. where $n \geq 6 \times 10^7$ in the case of city dataset), this $O(n^2)$ algorithm is too slow. To overcome this, we apply pruning techniques. Specifically, instead of considering all $i \in \{0,\dots,n\}$, we only consider $i \in [pn-\tau, pn+\tau]$ for some small number τ. Since we are only dropping non-negative terms, this still gives us a valid lower bound on δ. Furthermore, due to standard concentration bounds, it can easily be seen that taking $\tau = O(\sqrt{n} \cdot \mathrm{poly}\log(n/\delta))$ suffices to compute the sum (10) to within an error of say 0.0001δ. By applying a similar prunning to j, we end up with an algorithm that runs in time $O(n \cdot \mathrm{poly}\log(n/\delta))$, which suffices for our purposes.

G.4. B-Randomized Response and RAPPOR

For B-RR, it is simple to see that the mechanism becomes more private as p increases for any $p \in [0,1]$. The same holds for RAPPOR for $p \in [0,1/2]$. Once again, we use a similar approach as in the previous subsection to give optimistic estimates for the noise parameters of B-RR and RAPPOR. As discussed in Appendix E, even these optimistic errors are already noticably larger than those of the other protocols considered.

Recall form the previous subsection that we only need to provide the following subroutine: given p,ε, find a lower bound on δ. We will describe a generic algorithm for such a task in the next two subsections. After that, we will describe how to initiate the algorithms specifically for B-Randomized Response and RAPPOR.

G.4.1. Lower Bound on δ for Single Message Randomizers

In this subsection, we give a generic lower bound on δ given ε for any single-message mechanism $\mathcal{M}$ in the shuffled model. Suppose that the set of input of each user is $[B]$. For every $b \in [B]$, we use $\mathcal{D}_b$ to denote the distribution of the output message of the randomizer when the input is b.

To derive this lower bound, we consider two neighboring databases $\mathbf{x} = (1,\cdots,1,x)$ and $\mathbf{x}' = (1,\cdots,1,x')$ where $x \neq x'$ are from $\{2,\dots,B\}$. We then compute the ε-hockey stick divergence of $\mathcal{M}(\mathbf{x})$ and $\mathcal{M}(\mathbf{x}')$. Due to Lemma 20, this is a lower bound on δ for which $\mathcal{M}$ is (ε,δ)-DP.

An advantage in taking vectors $\mathbf{x},\mathbf{x}'$ as specified above is that their hockey stick divergence turns out to have a reasonably simple formula:

Lemma 36. *For $\mathbf{x} = (1,\dots,1,x)$ and $\mathbf{x}' = (1,\dots,1,x')$, we have*

$$d_\varepsilon(\mathcal{M}(\mathbf{x})||\mathcal{M}(\mathbf{x}')) = \frac{1}{n}\mathbb{E}\left[\left[\sum_{i=1}^{n} U_i\right]_+\right], \quad (12)$$

where $U_1,\dots,U_n$ are i.i.d. random variables where U_i is sampled as follows. Sample an outcome (i.e. a possible output message) o according to the distribution $\mathcal{D}_1$ and let

$$U_i = \frac{\mathcal{D}_x(o) - e^\varepsilon \cdot \mathcal{D}_{x'}(o)}{\mathcal{D}_1(o)}.$$

The proof of Lemma 36 is essentially the same as that of Lemma 5.3 from (Balle et al., 2019), except that we replace the "blanket distribution" there with the output distribution with 1 as an input $\mathcal{D}_1$. Nonetheless, we give the full proof for completeness below.

Proof of Lemma 36. Throughout the proof, we use $\mathbf{1}$ to denote the all-zero vector.

Consider any multiset of messages $\mathbf{o} = \{o_1,\dots,o_n\}$. The probability that this multiset $\mathbf{o}$ is the output of $\mathcal{M}(\mathbf{x})$ can be rearranged as follows:

$$\Pr[\mathcal{M}(\mathbf{x}) = \mathbf{o}]$$
$$= \sum_{\pi:[n]\to[n]} \mathcal{D}_1(o_{\pi(1)}) \cdots \mathcal{D}_1(o_{\pi(n-1)})\mathcal{D}_x(o_{\pi(n)})$$

$$= \sum_{\pi:[n]\to[n]} \mathcal{D}_1(o_1)\cdots\mathcal{D}_1(o_n)\frac{\mathcal{D}_x(o_{\pi(n)})}{\mathcal{D}_1(o_{\pi(n)})}$$
$$= n!\cdot\mathcal{D}_1(o_1)\cdots\mathcal{D}_1(o_n)\cdot\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\mathcal{D}_x(o_i)}{\mathcal{D}_1(o_i)}\right)$$
$$= \Pr[\mathcal{M}(\mathbf{1})=\mathbf{o}]\cdot\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\mathcal{D}_x(o_i)}{\mathcal{D}_1(o_i)}\right).$$

Similarly, we have

$$\Pr[\mathcal{M}(\mathbf{x}')=\mathbf{o}] = \Pr[\mathcal{M}(\mathbf{1})=\mathbf{o}]\cdot\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\mathcal{D}_{x'}(o_i)}{\mathcal{D}_1(o_i)}\right).$$

As a result,

$$\begin{aligned}
&d_\varepsilon(\mathcal{M}(\mathbf{x})||\mathcal{M}(\mathbf{x}'))\\
&= \sum_{\mathbf{o}}\left[\Pr[\mathcal{M}(\mathbf{x})=\mathbf{o}] - e^\varepsilon\cdot\Pr[\mathcal{M}(\mathbf{x}')=\mathbf{o}]\right]_+\\
&= \sum_{\mathbf{o}}\frac{\Pr[\mathcal{M}(\mathbf{1})=\mathbf{o}]}{n}\cdot\left[\sum_{i=1}^{n}\frac{\mathcal{D}_x(o_i)-e^\varepsilon\cdot\mathcal{D}_{x'}(o_i)}{\mathcal{D}_1(o_i)}\right]_+\\
&= \mathbb{E}_{\mathbf{o}\sim\mathcal{M}(\mathbf{1})}\left[\sum_{\mathbf{o}}\frac{1}{n}\cdot\left[\sum_{i=1}^{n}\frac{\mathcal{D}_x(o_i)-e^\varepsilon\cdot\mathcal{D}_{x'}(o_i)}{\mathcal{D}_1(o_i)}\right]_+\right]\\
&= \frac{1}{n}\mathbb{E}_{o_1,\cdots,o_n\sim\mathcal{D}_1}\left[\left[\sum_{i=1}^{n}\frac{\mathcal{D}_x(o_i)-e^\varepsilon\cdot\mathcal{D}_{x'}(o_i)}{\mathcal{D}_1(o_i)}\right]_+\right]\\
&= \frac{1}{n}\mathbb{E}\left[\left[\sum_{i=1}^{n}U_i\right]_+\right].
\end{aligned}$$

□

G.4.2. Efficiently Approximating Expected Positive Part of Sum of I.I.D. Random Variables

Our task now becomes how to compute the right hand side term of (12). While this seems hard in general, the U_i's that we will use below have very specific structures. Specifically, its support is only of size three; that is,

$$U_i = \begin{cases} d_1 & \text{with probability } p_1,\\ d_2 & \text{with probability } p_2,\\ d_3 & \text{with probability } p_3.\end{cases}$$

It turns out that, in this case, the problem is trivially solvable in $O(n^2)$ time, because the desired non-negative part of sum $\mathbb{E}\left[\left[\sum_{i=1}^{n}U_i\right]_+\right]$ can be written in the following form:

$$\sum_{a_1=0}^{n}\mathrm{Bin}(a_1;n,p_1)\cdot\sum_{a_2=0}^{n-a_1}\mathrm{Bin}\left(a_2;n-a_1,\frac{p_2}{p_2+p_3}\right)\cdot[a_1d_1+a_2d_2+(n-a_1-a_2)d_3]_+.$$

Although the trivial computation of the above expression requires $O(n^2)$ time, we can apply pruning techniques similar to those in Section G.3 to reduce the running time to $O(n\cdot\mathrm{poly}\log(n/\delta))$, which suffices for our purposes.

G.4.3. Parameters for B-Randomized Response

For B-RR, we do not use Lemma 36 directly on the messages, but we first group the messages into three groups (1) the message x, (2) the message x' and (3) all other messages. In other words, we "merge" all messages that are neither x nor x' into a single outcome. Since merging messages does not increase[12] hockey-stick divergence, we may compute the hockey stick divergence between $\mathcal{M}(\mathbf{x})$ and $\mathcal{M}(\mathbf{x}')$ after such a merging to get a lower bound on δ. In this case, it is not hard to check that we have

$$U_i = \begin{cases} \frac{B}{p}\left((1-p+\frac{p}{B})-e^\varepsilon\cdot\frac{p}{B}\right) & \text{with probability } \frac{p}{B},\\ \frac{B}{p}\left(\frac{p}{B}-e^\varepsilon\cdot(1-p+\frac{p}{B})\right) & \text{with probability } \frac{p}{B},\\ \frac{p(B-2)}{B-2p}\cdot(1-e^\varepsilon) & \text{w.p. } 1-\frac{2p}{B},\end{cases}$$

where the first value corresponds to when the output is x, the second to when the output is x', and the third to when the output is neither x nor x'.

We use the algorithm from the previous subsection to compute a lower bound on $\mathbb{E}\left[\left[\sum_{i=1}^{n}U_i\right]_+\right]$, which from Lemma 36 gives us a lower bound on δ.

G.4.4. Parameters for RAPPOR

Similarly, for RAPPOR, we first group the messages into three types: (i) the string has the xth bit set to one, the x'th bit set to zero and the first bit set to zero, (ii) the string has the x'th bit set to one, the xth bit set to zero and the first bit set to zero, and (ii) all other messages. We remark here that the grouping here is necessary as otherwise there are 7 possible values that the random variable U_i can take, which would result in an algorithm that is too slow (even after pruning). With this grouping, it is simple to check that our U_i has the following distribution:

$$U_i = \begin{cases} \left(\frac{1-p}{p}\right)^2 - e^\varepsilon & \text{w.p. } p^2(1-p),\\ 1-e^\varepsilon\left(\frac{1-p}{p}\right)^2 & \text{w.p. } p^2(1-p),\\ (1-e^\varepsilon)\cdot\frac{1-(1-p)^3-p^2(1-p)}{1-2p^2(1-p)} & \text{w.p. } 1-2p^2(1-p),\end{cases}$$

where the three values corresponding to the three groups respectively.

Once again, we then use the algorithm from Subsection G.4.2 to compute a lower bound on $\mathbb{E}\left[\left[\sum_{i=1}^{n}U_i\right]_+\right]$, which from Lemma 36 gives us a lower bound on δ.

[12] This is because such a merging is a form of post-processing.