Pure-DP Aggregation in the Shuffle Model: Error-Optimal and Communication-Efficient

Badih Ghazi
Google Research, Mountain View, CA, USA

Ravi Kumar
Google Research, Mountain View, CA, USA

Pasin Manurangsi
Google Research, Bangkok, Thailand

Abstract

We obtain a new protocol for binary counting in the ϵ-DP$_{\text{shuffle}}$ model with error $O(1/\epsilon)$ and expected communication $\widetilde{O}\left(\frac{\log n}{\epsilon}\right)$ messages per user. Previous protocols incur either an error of $O(1/\epsilon^{1.5})$ with $O_\epsilon(\log n)$ messages per user (Ghazi et al., ITC 2020) or an error of $O(1/\epsilon)$ with $O_\epsilon(n^2)$ messages per user (Cheu and Yan, TPDP 2022). Using the new protocol, we obtained improved ϵ-DP$_{\text{shuffle}}$ protocols for real summation and histograms.

2012 ACM Subject Classification Security and privacy; Security and privacy → Information-theoretic techniques

Keywords and phrases Differential Privacy, Shuffle Model, Aggregation, Pure Differential Privacy

Digital Object Identifier 10.4230/LIPIcs.ITC.2024.4

1 Introduction

Differential privacy (DP) [11] is a widely accepted notion used for bounding and quantifying an algorithm's leakage of personal information. Its most basic form, known as *pure*-DP, is governed by a single parameter $\epsilon > 0$, which bounds the leakage of the algorithm. Specifically, a randomized algorithm $A(\cdot)$ is said to be *ϵ-DP* if for any subset S of output values, and for any two datasets D and D' differing on a single user's data, it holds that $\Pr[A(D) \in S] \leq e^\epsilon \cdot \Pr[A(D') \in S]$. In settings where pure-DP is not (known to be) possible, a common relaxation is the so-called *approximate*-DP [10], which has an additional parameter $\delta \in [0, 1]$. In this case, the condition becomes: $\Pr[A(D) \in S] \leq e^\epsilon \cdot \Pr[A(D') \in S] + \delta$. Understanding the gap between pure- and approximate-DP algorithms is a natural and fundamental question that has been studied for a variety of analytics tasks (e.g., [23, 6]). Besides this, pure-DP protocols might be preferable in practice since an approximate-DP protocol may allow a (very small) non-zero probability of a catastrophic event, e.g., that the entire database is leaked[1].

Depending on the trust assumptions, three models of DP are commonly studied. The first is the *central* model, where a trusted curator is assumed to hold the raw data and is required to release a private output; this goes back to the first work of Dwork et al. [11] on DP. The second is the *local* model [13, 11, 22], where each user's message is required to be private. The third is the *shuffle* model [5, 8, 12], where the users' messages are routed through a trusted shuffler, which is assumed to be non-colluding with the curator, and which is expected to randomly permute the messages incoming from the different users (DP$_{\text{shuffle}}$). Formally, a protocol $P = (R, S, A)$ in the shuffle model consists of three procedures: (i)

[1] Indeed, such a catastrophic event can happen in some approximate-DP$_{\text{shuffle}}$ protocols proposed in previous works [18, 19].

a local randomizer $R(\cdot)$ that takes as input the data of a single user and outputs one or more messages, (ii) a shuffler $S(\cdot)$ that randomly permutes the messages from all the local randomizers, and (iii) an analyst $A(\cdot)$ that consumes the permuted output of the shuffler; the output of the protocol P is the output of the analyst $A(\cdot)$. Privacy in the shuffle model is defined as follows:

▶ **Definition 1** ([8, 12]). *A protocol $P = (R, S, A)$ is said to be (ϵ, δ)-$\mathrm{DP}_{\mathrm{shuffle}}$ if for any input dataset $D = (x_1, \ldots, x_n)$ where n is the number of users, it holds that $S(R(x_1), \ldots, R(x_n))$ is (ϵ, δ)-DP. In the case where $\delta = 0$, the protocol P is said to be ϵ-$\mathrm{DP}_{\mathrm{shuffle}}$.*

For several analytics tasks, low-error algorithms are known in the central model, whereas such algorithms are known to be impossible in the local model. For these analytic tasks, low-error algorithms are commonly sought in the shuffle model, since it is more preferable to trust a shuffler than a central curator. We note that while in this paper we treat the shuffler as a black box, multiple possible implementations have been considered in the literature including via secure hardware, mixnets and lightweight cryptographic protocols; see, e.g., the discussion in [5].

Interestingly, almost all algorithms studied in the shuffle model are for the approximate-DP setting. The only exceptions, to the best of our knowledge, are the pure-DP algorithms of Ghazi et al. [15] and Cheu and Yan [9] for binary summation; we discuss these next.

1.1 Our Contributions

In the *binary summation* (aka *counting*) problem, each user i receives an input $x_i \in \{0, 1\}$ and the goal is to estimate $\sum_{i \in [n]} x_i$. For this well-studied task, the discrete Laplace mechanism is known to achieve the optimal (expected absolute) error of $O(1/\epsilon)$ for ϵ-DP summation in the central model [21, 14]. Note that this error is independent of the number n of users, and is an absolute constant for the common parameter regime where ϵ is a constant. In contrast, the error of any aggregation protocol in the local model is known to be at least on the order of $\sqrt{n}$ [4, 7]. There have been many works that studied aggregation in the $\mathrm{DP}_{\mathrm{shuffle}}$ setting including [2, 3, 20, 15, 18, 19, 17, 1]. For pure-DP aggregation, it is known that any single-message protocol (where each user sends a single message to the shuffler) should incur error $\Omega_\epsilon(\sqrt{n})$ [1]. For multi-message protocols, where each user can send multiple messages to the shuffler, the best known protocols incur either an error of $O(1/\epsilon^{1.5})$ with $O(\log n)$ messages per user [15] or an error of $O(1/\epsilon)$ with $O(n^2)$ messages per user [9]. No protocol simultaneously achieved error $O(1/\epsilon)$ and communication $O(\log n)$.

In this paper, we obtain an ϵ-$\mathrm{DP}_{\mathrm{shuffle}}$ algorithm for binary summation, where each user, in expectation, sends $O\left(\frac{\log n}{\epsilon}\right)$ one-bit messages; this answers the main open question for this basic aggregation task.

▶ **Theorem 2.** *For every positive real number $\epsilon \leq O(1)$, there is a (non-interactive) ϵ-$\mathrm{DP}_{\mathrm{shuffle}}$ protocol for binary summation with root mean square error $O(1/\epsilon)$, where each user sends $O\left(\frac{\log n}{\epsilon}\right)$ messages in expectation and each message consists of a single bit.*

In fact, similar to the protocol of Cheu and Yan [9], our protocol can get an error that is arbitrarily close to that of the discrete Laplace mechanism, which is known to be optimal in the central model for any $\epsilon > 0$; see [21, 14]. We defer the formal statement to Theorem 6.

Before we continue, we note that while the expected number of messages in Theorem 2 is small (and with an exponential tail), the *worst* case number of messages is unbounded. This should be contrasted with an $\Omega_\epsilon(\sqrt{\log n})$ lower bound in [15] that only applies to the worst case number of bits sent by a user. We discuss this further in Section 6.

Protocols for Real Summation and Histogram

Using known techniques (e.g., [8, 15]), we immediately get the following consequences for real summation and histogram.

In the *real summation* problem, each x_i is a real value in $[0, 1]$; the goal is again to estimate the sum $\sum_{i\in[n]} x_i$. The protocol in [15] achieves an expected root mean square error (RMSE) of $\widetilde{O}(1/\epsilon^{1.5})$; here, each user sends $O_\epsilon(\log^3 n)$ messages each of length $O(\log\log n)$ bits. By running their protocol bit-by-bit with an appropriate privacy budget split, we get an algorithm with an improved, and asymptotically optimal, error of $O(1/\epsilon)$ while with expected communication similar to theirs.

▶ **Corollary 3.** *For every positive real number $\epsilon \le O(1)$, there is a (non-interactive) ϵ-$\mathrm{DP}_{\mathrm{shuffle}}$ protocol for real summation with RMSE $O(1/\epsilon)$, where each user sends $O(\frac{\log^3 n}{\epsilon})$ messages in expectation and each message consists of $O(\log\log n)$ bits.*

A widely used primitive related, though not identical, to aggregation is histogram computation. In the *histogram* problem, each x_i is a number in $[B]$; the goal is to estimate the histogram of the dataset, where the histogram $\mathbf{h} \in \mathbb{Z}^B_{\ge 0}$ is defined by $h_b = |\{i \in [n] \mid x_i = b\}|$. The error of an estimated histogram $\tilde{\mathbf{h}}$ is usually measured in the ℓ_∞-sense, i.e., $\|\tilde{\mathbf{h}} - \mathbf{h}\|_\infty = \max_{b\in[B]} |h_b - \tilde{h}_b|$.

For this task, which has been studied in several papers including [16, 1], the best known pure-$\mathrm{DP}_{\mathrm{shuffle}}$ protocol achieved ℓ_∞-error $O\left(\frac{\log B \log n}{\epsilon^{1.5}}\right)$ and communication $O\left(\frac{B \log n \log B}{\epsilon}\right)$ bits. By running our $(\epsilon/2)$-$\mathrm{DP}_{\mathrm{shuffle}}$ protocol separately for each bucket [15, Appendix A], we immediately arrive at the following:

▶ **Corollary 4.** *For every positive real number $\epsilon \le O(1)$, there is a (non-interactive) ϵ-$\mathrm{DP}_{\mathrm{shuffle}}$ protocol that computes histograms on domains of size B with an expected ℓ_∞-error of at most $O\left(\frac{\log B \log n}{\epsilon}\right)$, where each user sends $O\left(\frac{B \log n}{\epsilon}\right)$ messages in expectation and each message consists of $O(\log B)$ bits.*

1.2 Technical Overview

We will now briefly discuss the proof of Theorem 2. Surprisingly, we show that a simple modification of the algorithm from [18] satisfies pure-DP! To understand the modification and its necessity, it is first important to understand their algorithm. In their protocol, the messages are either $+1$ or -1, and the analyzer's output is simply the sum of all messages. There are three type of messages each user sends:

- *Input-Dependent Messages*: If the input x_i is 1, the user sends a $+1$ message. Otherwise, the user does not send anything.
- *Flooding Messages*: These are messages that do *not* affect the final estimation error. In particular, a random variable $z_i^{\pm 1}$ is drawn from an appropriate distribution and the user sends $z_i^{\pm 1}$ additional copies of -1 and $z_i^{\pm 1}$ additional copies of $+1$. These messages get canceled when the analyzer computes it output.
- *Noise Messages*: These are the messages that affect the error in the end. Specifically, z_i^{+1}, z_i^{-1} are drawn independently from an appropriate distribution, and z_i^{-1} additional copies of -1 and z_i^{+1} additional copies of $+1$ are then sent.

We note here that the view of the analyzer is simply the number of $+1$ messages and the number of -1 messages, which we will denote by V_{+1} and V_{-1} respectively.

While [18] show that this protocol is (ϵ, δ)-DP, it is easy to show that this is *not* ϵ-DP for any finite ϵ. Indeed, consider two neighboring datasets where X consists of all zeros and X'

consists of a single one and $n-1$ zeros. There is a non-zero probability that $V_{+1}(X)=0$, while $V_{+1}(X')$ is always non-zero (because of the input-dependent message from the user holding the single one).

To fix this, we randomize this "input-dependent" part. With probability q, the user sends nothing. With the remaining probability $1-q$, (instead of sending a single $+1$ for $x_i=1$ as in [18],) the user sends $s+1$ copies of $+1$ and s copies of -1; similarly, for $x_i=0$, the user sends s copies of $+1$ and s copies of -1. By setting q to be sufficiently small (e.g., $q=O(1/\epsilon n)$), it can be shown that the error remains roughly the same as before. Furthermore, when s is sufficiently large (i.e., $O_\epsilon(\log n)$), we manage to show that this algorithm satisfies ϵ-$\mathrm{DP}_{\mathrm{shuffle}}$. While the exact reason for this pure-DP guarantee is rather technical, the general idea is similar to [15]: by making the "border" part of the support equal in probabilities in the two cases, we avoid the issues presented above. Furthermore, by making s sufficiently large, the input-dependent probability is "sufficiently inside" of the support that it usually does not completely dominate the contribution from the outer part.

Finally, note that V_{+1}, V_{-1} involves summation of many i.i.d. random variables $\sum_{i\in[n]} z_i^{\pm 1}$, $\sum_{i\in[n]} z_i^{+1}$, and $\sum_{i\in[n]} z_i^{-1}$. As observed in [18], it is convenient to use *infinitely divisible* distributions so that these sums have distributions that are independent of n, allowing for simpler calculations. We inherit this feature from their analysis.

2 Preliminaries

For a discrete distribution $\mathcal{D}$, let $f_{\mathcal{D}}$ denote its probability mass function (PMF). The *max-divergence* between distributions $\mathcal{D}_1, \mathcal{D}_2$ is defined as $d_\infty(\mathcal{D}_1\|\mathcal{D}_2) := \max_{x\in\mathrm{supp}(\mathcal{D}_1)} \ln \frac{f_{\mathcal{D}_1}(x)}{f_{\mathcal{D}_2}(x)}$.

For two distributions $\mathcal{D}_1, \mathcal{D}_2$ over $\mathbb{Z}^d$, we write $\mathcal{D}_1 * \mathcal{D}_2$ to denote its *convolution*, i.e., the distribution of z_1+z_2 where $z_1\sim\mathcal{D}_1, z_2\sim\mathcal{D}_2$ are independent. Moreover, let $(\mathcal{D})^{*n}$ denote the n-fold convolution of $\mathcal{D}$, i.e., the distribution of $z_1+\cdots+z_n$ where $z_1,\ldots,z_n\sim\mathcal{D}$ are independent. We write $\mathcal{D}\otimes\mathcal{D}'$ to denote the *product* distribution of $\mathcal{D}_1, \mathcal{D}_2$. Furthermore, we may write a value to denote the distribution all of whose probability mass is at that value (e.g., 0 stands for the probability distribution that is always equal to zero).

A distribution $\mathcal{D}$ is *infinitely divisible* iff, for every positive integer n, there exists a distribution $\mathcal{D}_{/n}$ such that $\mathcal{D} = (\mathcal{D}_{/n})^{*n}$. Two distributions we will use here (both supported on $\mathbb{Z}_{\geq 0}$) are:

- *Poisson Distribution* $\mathrm{Poi}(\lambda)$: This is the distribution whose PMF is $f_{\mathrm{Poi}(\lambda)}(k) = \lambda^k e^{-\lambda}/k!$. It satisfies $\mathrm{Poi}(\lambda)_{/n} = \mathrm{Poi}(\lambda/n)$.
- *Negative Binomial Distribution* $\mathrm{NB}(r,p)$: Its PMF is $f_{\mathrm{NB}(r,p)}(k) = \binom{k+r-1}{k} p^r (1-p)^k$. It satisfies $\mathrm{NB}(r,p)_{/n} = \mathrm{NB}(r/n, p)$.
  - *Geometric Distribution* $\mathrm{Geo}(p)$: A special case of the NB distribution is the geometric distribution $\mathrm{Geo}(p) = \mathrm{NB}(1,p)$, i.e., one with $f_{\mathrm{Geo}(p)}(k) = p(1-p)^k$.

Finally, we recall that the *discrete Laplace distribution* $\mathrm{DLap}(a)$ is a distribution supported on $\mathbb{Z}$ with PMF $f_{\mathrm{DLap}(a)}(x) \propto \exp(-a|x|)$. It is well-known that $\mathrm{DLap}(a)$ is the distribution of $z_1 - z_2$ where $z_1, z_2 \sim \mathrm{Geo}(1-\exp(-a))$ are independent. Furthermore, the variance of the discrete Laplace distribution is $\mathrm{Var}(\mathrm{DLap}(a)) = \frac{2e^{-a}}{(1-e^{-a})^2}$.

We will also use the following well-known lemma[2]:

▶ **Lemma 5.** *For any distributions $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ over $\mathbb{Z}^d$, $d_\infty(\mathcal{D}_1 * \mathcal{D}_3 \| \mathcal{D}_2 * \mathcal{D}_3) \leq d_\infty(\mathcal{D}_1\|\mathcal{D}_2)$.*

[2] This can be viewed as a special case of the post-processing property of DP where the post-processing function is adding a random variable drawn from $\mathcal{D}_3$. Another way to see that this holds is to simply observe that, for any $y\in\mathrm{supp}(\mathcal{D}_1*\mathcal{D}_2)$, we have $f_{\mathcal{D}_1*\mathcal{D}_3}(y) = \sum_{z\in\mathrm{supp}(\mathcal{D}_3)} f_{\mathcal{D}_3}(z)\cdot f_{\mathcal{D}_1}(y-z) \leq \sum_{z\in\mathrm{supp}(\mathcal{D}_3)} f_{\mathcal{D}_3}(z)\cdot\left(e^{d_\infty(\mathcal{D}_1\|\mathcal{D}_2)}\cdot f_{\mathcal{D}_2}(y-z)\right) = e^{d_\infty(\mathcal{D}_1\|\mathcal{D}_2)} f_{\mathcal{D}_2*\mathcal{D}_3}(y)$.

3 Counting Protocol

In this section, we will describe a pure-$\mathrm{DP}_{\mathrm{shuffle}}$ algorithm for counting, which is our main result.

▶ **Theorem 6.** *For any positive real numbers $\epsilon \leq O(1)$ and $\rho \in (0, 1/2]$, there is a (non-interactive) ϵ-$\mathrm{DP}_{\mathrm{shuffle}}$ protocol for binary summation that has MSE at most $(1+\rho) \cdot \mathrm{Var}(\mathrm{DLap}(\epsilon))$ where each user sends $O\left(\frac{\log(n/\rho)}{\epsilon\rho}\right)$ messages in expectation and each message consists of a single bit.*

By setting ρ arbitrarily close to zero, we can get the mean-square error (MSE) to be arbitrarily close to that of the discrete Laplace mechanism, which is known to be (asymptotically) optimal in the central model [21, 14]. We can get this guarantee for other type of errors, e.g., ℓ_1-error (aka expected absolute error) as well, but for ease of presentation, we only focus on the MSE.

Note that Theorem 6 implies Theorem 2 by simply setting ρ to be a positive constant (say, 0.5).

3.1 Algorithm

In this section we present and analyze our main algorithm for counting (aka binary summation). To begin, we will set our parameters as follows.

▶ **Condition 7.** *Let $\lambda, \epsilon', \epsilon, q \in \mathbb{R}_{>0}$ and $s \in \mathbb{Z}_{>0}$. Suppose that the following conditions hold:*

- $\epsilon' < \epsilon$,
- $s \geq 2 \ln\left(\frac{1}{(e^{\epsilon}-1)q}\right) / (\epsilon - \epsilon')$,
- $\lambda \geq \frac{e^{\epsilon-\epsilon'}}{e^{(\epsilon-\epsilon')/2}-1} \cdot s$.

We now define the following distributions:

- $\mathcal{D}^{\mathrm{noise}} = \mathrm{Geo}(1 - e^{-\epsilon'})$.
- $\mathcal{D}^{\mathrm{flood}} = \mathrm{Poi}(\lambda)$.
- For $x \in \{0, 1\}$, $\mathcal{D}^{\mathrm{input},x}$ supported on $\mathbb{Z}^2_{\geq 0}$ is defined as

$$\mathcal{D}^{\mathrm{input},x}((s+x, s)) = 1 - q,$$
$$\mathcal{D}^{\mathrm{input},x}((0, 0)) = q.$$

Algorithm 1 contains the formal description of the randomizer and Algorithm 2 contains the description of the analyzer. As mentioned earlier, our algorithm is the same as that of [18], except in the first step (Line 2). In their work, the protocol always sends a single $+1$ if $x_i = 1$ and nothing otherwise in this step. Instead, we randomize this step by always sending nothing with a certain probability. With the remaining probability, instead of sending a single $+1$ for $x_i = 1$, we send $s + 1$ copies of $+1$ and s copies of -1 (similarly, we send s copies of $+1$ and s copies of -1 in the case $x_i = 0$).

4 Analysis of the Protocol

In this section we analyze the privacy, utility, and communication guarantees of our counting protocol. Throughout the remainder of this section, we assume the distributions and parameters are set as in Condition 7; for brevity, we will not state this assumption in our privacy analysis.

Algorithm 1 Counting Randomizer.

procedure $\textsc{CorrNoiseRandomizer}_n(x_i)$
  Sample $(y_i^{+1}, y_i^{-1}) \sim \mathcal{D}^{\text{input},x_i}$
  Sample $z_i^{+1}, z_i^{-1} \sim \mathcal{D}^{\text{noise}}_{/n}$
  Sample $z_i^{\pm 1} \sim \mathcal{D}^{\text{flood}}_{/n}$
  Send $y_i^{+1} + z_i^{+1} + z_i^{\pm 1}$ copies of $+1$, and $y_i^{-1} + z_i^{-1} + z_i^{\pm 1}$ copies of -1

Algorithm 2 Counting Analyzer.

procedure $\textsc{CorrNoiseAnalyzer}_q$
  $R \leftarrow$ multiset of messages received
  return $\frac{1}{1-q}\left(\sum_{y \in R} y\right)$

4.1 Privacy Analysis

▶ **Lemma 8** (Main Privacy Guarantee). *CorrNoiseRandomizer satisfies* ϵ-$\text{DP}_{\text{shuffle}}$.

To prove the above, we need the following technical lemmas regarding $\mathcal{D}^{\text{noise}}, \mathcal{D}^{\text{flood}}$.

▶ **Lemma 9.** *For every* $i \in \mathbb{Z}$, $f_{\mathcal{D}^{\text{noise}}}(i-1) \le e^{\epsilon'} f_{\mathcal{D}^{\text{noise}}}(i)$

Proof. This immediately follows from the PMF definition of $\mathcal{D}^{\text{noise}} = \text{Geo}(1 - e^{\epsilon'})$. ◀

▶ **Lemma 10.** *For every* $i \in \mathbb{Z}$, $(e^{\epsilon} - 1)q \cdot f_{\mathcal{D}^{\text{flood}}}(i+s) + e^{\epsilon-\epsilon'} f_{\mathcal{D}^{\text{flood}}}(i-1) \ge f_{\mathcal{D}^{\text{flood}}}(i)$.

Proof. If $e^{\epsilon-\epsilon'} f_{\mathcal{D}^{\text{flood}}}(i-1) \ge f_{\mathcal{D}^{\text{flood}}}(i)$, then the statement is clearly true. Otherwise, we have $f_{\mathcal{D}^{\text{flood}}}(i) > 0$ (i.e., $i \ge 0$) and $e^{\epsilon'-\epsilon} > \frac{f_{\mathcal{D}^{\text{flood}}}(i-1)}{f_{\mathcal{D}^{\text{flood}}}(i)} = \frac{i}{\lambda}$, which implies

$$0 \le i \le e^{\epsilon'-\epsilon}\lambda. \tag{1}$$

We can then bound $\frac{f_{\mathcal{D}^{\text{flood}}}(i+s)}{f_{\mathcal{D}^{\text{flood}}}(i)}$ as

$$\begin{aligned}\frac{f_{\mathcal{D}^{\text{flood}}}(i+s)}{f_{\mathcal{D}^{\text{flood}}}(i)} &= \frac{\lambda^s}{(i+1)\cdots(i+s)} \ge \frac{\lambda^s}{(i+s)^s} \\ &\overset{(1)}{\ge} \left(\frac{\lambda}{e^{\epsilon'-\epsilon}\lambda + s}\right)^s \ge \left(\frac{\lambda}{e^{(\epsilon'-\epsilon)/2}\lambda}\right)^s \\ &\ge \frac{1}{(e^{\epsilon}-1)q},\end{aligned}$$

where the last two inequalities follow from our assumptions on λ and s respectively (Condition 7). Thus, in this case, we also have $(e^{\epsilon}-1)q \cdot f_{\mathcal{D}^{\text{flood}}}(i+s) + e^{\epsilon-\epsilon'} f_{\mathcal{D}^{\text{flood}}}(i-1) \ge f_{\mathcal{D}^{\text{flood}}}(i)$ as desired. ◀

We are now ready to prove the privacy guarantee (Lemma 8).

Proof of Lemma 8. For any input dataset X. Let $V(X) = (V_{+1}, V_{-1})$ denote the distribution of the view of shuffler, where V_{+1} and V_{-1} denotes the number of $+1$ messages and the number of -1 messages respectively.

Consider two neighboring datasets $X = (x_1, \ldots, x_n)$ and $X' = (x'_1, \ldots, x'_n)$. Assume w.l.o.g. that they differ in the first coordinate and $x_1 = 1, x'_1 = 0$ and $x'_2 = x_2$, …, $x'_n = x_n$. To prove that CorrNoiseRandomizer satisfies ϵ-$\text{DP}_{\text{shuffle}}$, we need to prove that $d_\infty(V(X) \| V(X')) \le \epsilon$ and $d_\infty(V(X') \| V(X)) \le \epsilon$.

Let $\mathcal{F}$ denote the distribution on $\mathbb{Z}^2$ of (X, X) where $X \sim \mathcal{D}^{\text{flood}}$. Observe that

$$V(X) = \mathcal{D}^{\text{input},1} * \mathcal{D}^{\text{input},x_2} * \cdots * \mathcal{D}^{\text{input},x_n} * \mathcal{F} * (\mathcal{D}^{\text{noise}} \otimes 0) * (0 \otimes \mathcal{D}^{\text{noise}}),$$

and

$$V(X') = \mathcal{D}^{\text{input},0} * \mathcal{D}^{\text{input},x_2} * \cdots * \mathcal{D}^{\text{input},x_n} * \mathcal{F} * (\mathcal{D}^{\text{noise}} \otimes 0) * (0 \otimes \mathcal{D}^{\text{noise}}).$$

Bounding $d_\infty(V(X)\|V(X'))$

From Lemma 5, we have

$$d_\infty(V(X)\|V(X')) \le d_\infty(\mathcal{D}^{\text{input},1} * (\mathcal{D}^{\text{noise}} \otimes 0)\|\mathcal{D}^{\text{input},0} * (\mathcal{D}^{\text{noise}} \otimes 0)).$$

For any $i, j \in \mathbb{Z}$, we have

$$\begin{aligned}
&f_{\mathcal{D}^{\text{input},1} * \mathcal{D}^{\text{noise}} \otimes 0}(i,j) \\
&= q \cdot f_{\mathcal{D}^{\text{noise}}}(i)\mathbf{1}[j=0] + (1-q) \cdot f_{\mathcal{D}^{\text{noise}}}(i-s-1)\mathbf{1}[j=s] \\
&\le q \cdot f_{\mathcal{D}^{\text{noise}}}(i)\mathbf{1}[j=0] + (1-q) \cdot e^{\epsilon'} f_{\mathcal{D}^{\text{noise}}}(i-s)\mathbf{1}[j=s] \\
&\le e^{\epsilon}\left(q \cdot f_{\mathcal{D}^{\text{noise}}}(i)\mathbf{1}[j=0] + (1-q) \cdot f_{\mathcal{D}^{\text{noise}}}(i-s)\mathbf{1}[j=s]\right) \\
&= e^{\epsilon} \cdot f_{\mathcal{D}^{\text{input},0} * \mathcal{D}^{\text{noise}} \otimes 0}(i,j),
\end{aligned}$$

where the first inequality follows from Lemma 9 and the second inequality follows from Condition 7. Combining the above inequalities, we have $d_\infty(V(X)\|V(X')) \le \epsilon$ as desired.

Bounding $d_\infty(V(X')\|V(X))$

Again, from Lemma 5, we have

$$d_\infty(V(X')\|V(X)) \le d_\infty\left(\mathcal{D}^{\text{input},0} * \mathcal{F} * (0 \times \mathcal{D}^{\text{noise}}) \| \mathcal{D}^{\text{input},1} * \mathcal{F} * (0 \times \mathcal{D}^{\text{noise}})\right).$$

For any $i, j \in \mathbb{Z}$, we have

$$\begin{aligned}
&f_{\mathcal{D}^{\text{input},0} * \mathcal{F} * (0 \times \mathcal{D}^{\text{noise}})}(i,j) \\
&= f_{\mathcal{D}^{\text{input},0} * \mathcal{F}}(i,i) \cdot f_{\mathcal{D}^{\text{noise}}}(j-i) \\
&= (q \cdot f_{\mathcal{D}^{\text{flood}}}(i) + (1-q) \cdot f_{\mathcal{D}^{\text{flood}}}(i-s)) \cdot f_{\mathcal{D}^{\text{noise}}}(j-i) \\
&\le e^{\epsilon}\left(q \cdot f_{\mathcal{D}^{\text{flood}}}(i) + (1-q) \cdot e^{-\epsilon'} f_{\mathcal{D}^{\text{flood}}}(i-s-1)\right) \\
&\qquad \cdot f_{\mathcal{D}^{\text{noise}}}(j-i) \\
&\le e^{\epsilon}\left(q \cdot f_{\mathcal{D}^{\text{flood}}}(i) \cdot f_{\mathcal{D}^{\text{noise}}}(j-i)\right. \\
&\qquad \left. + (1-q) \cdot f_{\mathcal{D}^{\text{flood}}}(i-s-1) \cdot f_{\mathcal{D}^{\text{noise}}}(j-i+1)\right) \\
&= e^{\epsilon} f_{\mathcal{D}^{\text{input},1} * \mathcal{F} * (0 \times \mathcal{D}^{\text{noise}})}(i,j),
\end{aligned}$$

where the first inequality follows from Lemma 10 and the second inequality follows from Lemma 9. Combining the above two inequalities, we have $d_\infty(V(X')\|V(X)) \le \epsilon$, concluding our proof. ◀

4.2 Utility Analysis

We next analyze the MSE of the output estimate.

▶ **Lemma 11.** *The estimator from Algorithm 2 is unbiased and its MSE is at most*

$$\left(\frac{1}{1-q}\right)^2 \cdot (qn + \mathrm{Var}(\mathrm{DLap}(\epsilon'))) .$$

Proof. Notice that the output estimate is equal to

$$\frac{1}{1-q}\left(\sum_{i\in[n]}(y_i^{+1} - y_i^{-1} + z_i^{+1} - z_i^{-1})\right) = \frac{1}{1-q}\left(\sum_{i\in[n]}(y_i^{+1} - y_i^{-1}) + Z\right),$$

where $Z \sim \mathrm{DLap}(\epsilon')$. It is also simple to verify that $\mathbb{E}[y_i^{+1} - y_i^{-1}] = (1-q)x_i$. Thus, the estimator is unbiased as desired. Its MSE is equal to

$$\begin{aligned}&\mathrm{Var}\left(\frac{1}{1-q}\left(\sum_{i\in[n]}(y_i^{+1} - y_i^{-1}) + Z\right)\right)\\ &= \left(\frac{1}{1-q}\right)^2\left(\sum_{i\in[n]}\mathrm{Var}(y_i^{+1} - y_i^{-1}) + \mathrm{Var}(\mathrm{DLap}(\epsilon'))\right).\end{aligned}$$

Next, notice that, if $x_i = 0$, then $y_i^{+1} - y_i^{-1} - x_i = 0$ always. Otherwise, if $x_i = 1$, then $y_i^{+1} - y_i^{-1} - x_i = 0$ with probability $1-q$ and $y_i^{+1} - y_i^{-1} - x_i = 1$ with probability q. As a result, we have $\mathrm{Var}(y_i^{+1} - y_i^{-1}) \le q$. Plugging this into the above inequality yields the claimed bound on the MSE. ◀

4.3 Communication Analysis

The expected number of bits send by the users can be easily computed as follows.

▶ **Lemma 12.** *The expected number of messages sent by each user is at most* $2s + 1 + \frac{\lambda}{n} + O\left(\frac{1}{\epsilon' n}\right)$.

Proof. The expected number of bits sent per user is

$$\begin{aligned}&\mathbb{E}[y_i^{+1} + y_i^{-1}] + \mathbb{E}[z_i^{+1} + z_i^{-1}] + 2\mathbb{E}[z_i^{\pm 1}]\\ &\le (2s+1) + \frac{2\mathbb{E}[\mathcal{D}^{\mathrm{noise}}]}{n} + \frac{\mathbb{E}[\mathcal{D}^{\mathrm{flood}}]}{n}\\ &= 2s + 1 + O\left(\frac{1}{\epsilon' n}\right) + \frac{\lambda}{n}.\end{aligned}$$ ◀

4.4 Putting Things Together: Proof of Theorem 6

Finally, we are ready to prove Theorem 6 by plugging in appropriate parameters and invoke the previous lemmas.

Proof of Theorem 6. We start by picking $\epsilon' = \epsilon - 0.01\rho \cdot \min\{\epsilon, 1\}$. For this choice of ϵ', we have

$$\frac{\mathrm{Var}(\mathrm{DLap}(\epsilon'))}{\mathrm{Var}(\mathrm{DLap}(\epsilon))} = \frac{\frac{2e^{-\epsilon'}}{(1-e^{-\epsilon'})^2}}{\frac{2e^{-\epsilon}}{(1-e^{-\epsilon})^2}}$$

$$\leq 1 + \frac{(e^{\epsilon-\epsilon'}-1)(1+e^{-\epsilon'})}{1-e^{-\epsilon'}}$$
$$\leq 1 + \frac{3(\epsilon-\epsilon')\cdot 2}{\epsilon'} \leq 1 + 0.1\rho.$$

Then, picking

$$q = 0.1\rho \cdot \min\left\{\frac{\mathrm{Var}(\mathrm{DLap}(\epsilon))}{n}, 1\right\} = O\left(\frac{\rho}{\epsilon^2 n}\right),$$
$$s \geq 2\ln\left(\frac{1}{(e^{\epsilon}-1)q}\right)/(\epsilon-\epsilon') = O\left(\frac{\log(n/\rho)}{\epsilon\rho}\right),$$
$$\lambda \geq \frac{e^{\epsilon-\epsilon'}}{e^{(\epsilon-\epsilon')/2}-1}\cdot s = O\left(\frac{\log(n/\rho)}{\epsilon^2\rho}\right),$$

and applying Lemma 8, Lemma 11, and Lemma 12 immediately yields Theorem 6. (Note that we may assume that $\epsilon \geq 1/n$; otherwise we can just output zero. Under this assumption, we have $\lambda/n \leq O\left(\frac{\log(n/\rho)}{\epsilon\rho}\right)$ as desired for the communication complexity claim.) ◀

5 Non-Asymptotic Comparisons with Previous Work

In this section, we provide concrete non-asymptotic comparisons between our binary summation protocol and those from previous work [15, 9] for various population sizes n and privacy parameters ϵ. As we explain in more detail below, our results demonstrate that our protocol is much more practical than those of previous works.

First, we find that the parameters in the protocol of [15] are impractical; in fact, for $n \leq 800,000$, their protocol is *undefined* unless $\epsilon < 0.01$.[3] Furthermore, even in the regime that it is well-defined, their expected communication complexity is provably at least 1000x ours and their root-mean-square error (RMSE) is probably at least 100x ours. Hence, we only focus on the comparison between our algorithm and that of [9].

Parameter Setting

For both our algorithm and that of [9], one can achieve RMSE arbitrarily close to that of the ϵ-DP discrete Laplace mechanism in the central model. (See the parameter ρ in Theorem 6.) To reduce the parameter space for comparison, we set the parameters so that the RMSE of these protocols is within 10% of the discrete Laplace mechanism. Given this error target, we simply use the formulae from Lemma 11 and Condition 7 to optimize for ϵ', q, λ that minimizes the expected communication (according to Lemma 12); we use `scipy` package for this optimization. For [9]'s algorithm, we set the parameter in an optimistic manner so that we underestimate the communication required in their protocol[4].

[3] This is due to the fact that they require their parameter $p = \frac{100e^{100\epsilon}\log(1/(1-e^{0.1\epsilon}))}{n(1-e^{0.1\epsilon})}$ to be less than one. Of course, one can run their protocol at a smaller ϵ but this increases the communication and error even further.

[4] Namely, we only set p in their protocol to $0.5/n$ and do not account for the error from the p-probability event that the input is randomized. (In their analysis, p should actually be set to $\hat{q}/n$ where $\hat{q} \leq O(1/n)$ is yet another small parameter.)

Expected Communication Comparison

We provide a comparison of the expected number of messages sent when fixing $\epsilon = 1$ and varying n from 1 to 1000 in Figure 1(ii). To summarize, the number of messages of their protocols grows very quickly and exceed 10 million even when $n = 65$! This agrees with theory, which suggests that their communication complexity grows with $\widetilde{O}(n^2)$. Meanwhile, our protocol has expected number of messages sent less than 600 for the entire range of $10 < n \leq 100$, again agreeing with the theory that our communication grows only with $O\left(\frac{\log n}{\epsilon}\right)$. Moreover, the expected number of messages of our protocol is less than that of theirs except when $n = 1$. For clarity, we also provide our protocol's expected number of messages in Figure 2(i) for the small n case ($1 \leq n \leq 10^3$) and in Figure 2(ii) for the large n case ($10^3 \leq n \leq 10^6$). These plots show that the expected number of messages is large for very small $n \leq 5$, in which regime the expected number of messages decrease as n increases. This regime corresponds to the regime where the communication due to the Poisson noise dominates. Once the expected number of messages bottoms out, it increases slowly, as suggested by our theoretical analysis. Finally, we suspect that the curve is not completely smooth since `scipy.optimize.minimize_scalar` does not always find the optimum[5].

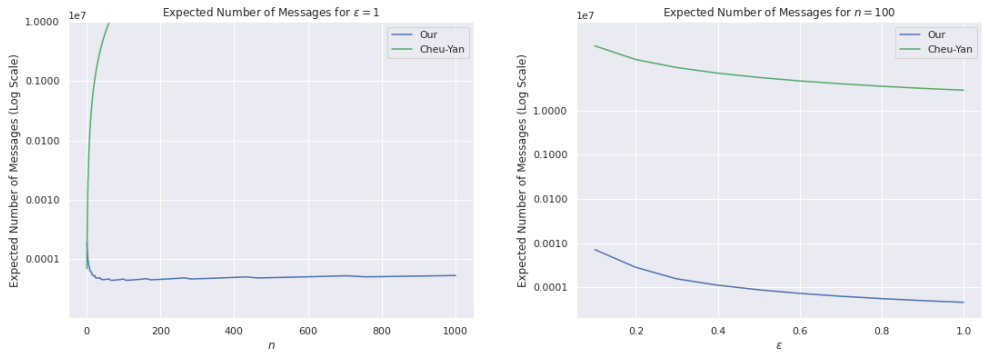

Figure 1 Comparison between the expected number of messages sent in our protocol and in Cheu–Yan protocol when (i) $\epsilon = 1$ and varying n, (ii) $n = 100$ and varying ϵ. (Note that the y-axis is in log-scale.)

Next, we fix $n = 100$ and vary ϵ. The resulting expected communication is presented in Figure 1(i). Again, there is a very large (> 10000x) gap between our expected communication and theirs. Furthermore, these increase roughly as $1/\epsilon$, as predicted by theory.

Finally, we note that, while we perform comparisons for binary summation, the comparisons would be similar for histogram as well. This is because all protocols are adapted to the histogram problem by simply running the binary protocol separately for each bucket; thus, the expected communication simply increases by a factor of B.

6 Conclusions and Open Questions

In this work, we have provided pure-DP$_{\text{shuffle}}$ algorithms that achieve nearly optimal errors for bit summation, real summation, and histogram while significantly improving on the communication complexity compared to the state-of-the-art. Despite this, there are still a number of interesting open questions, some of which we highlight below.

[5] In particular, the value of s is discrete in our optimization problem, making it harder to optimize for

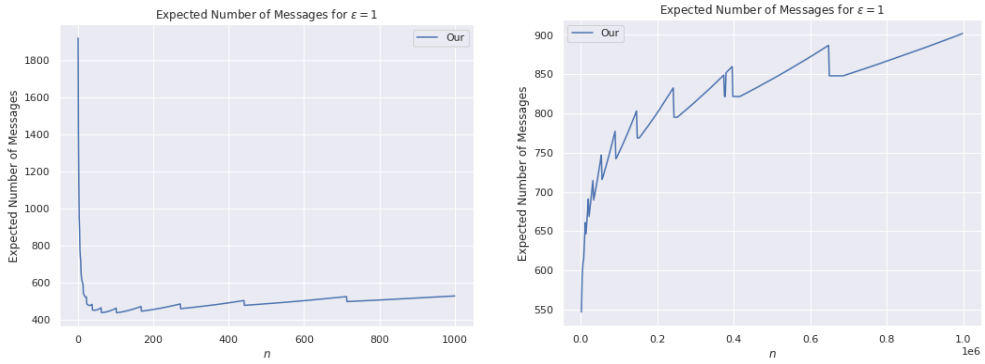

Figure 2 The expected number of messages sent in our protocol when $\epsilon = 1$ for (i) $1 \leq n \leq 10^3$, (ii) $10^3 \leq n \leq 10^6$.

- **Protocol with a bounded number of messages.** As mentioned briefly in Section 1.1, our protocol can result in an arbitrarily large number of messages per user, although the expected number is quite small. (In fact, the distribution of the number of messages enjoys a strong exponential tail bound.) Is it possible to design a pure-DP$_{\text{shuffle}}$ protocol where the maximum number of messages is $O\left(\frac{\log n}{\epsilon}\right)$ for binary summation?
  For this question, we note that a rather natural approach is to modify our protocol to make its number of messages bounded. Namely, we replace $\mathcal{D}^{\text{noise}}_{/n}$ and $\mathcal{D}^{\text{flood}}_{/n}$ by a truncated version of their respective distributions. It turns out that the latter is relatively simple (e.g., even replacing it with a Bernoulli distribution also works) because we only require a mild condition in Lemma 10 to hold. On the other hand, for the former, we are using Lemma 9, which only holds for unbounded distributions. We would like to stress that we do not know whether replacing $\mathcal{D}^{\text{noise}}_{/n}$ with a truncated version of the negative binomial distribution with a "symmetrized" the input dependent part[6] violates pure-DP; however, we do not know how to prove that it satisfies pure-DP either, as the probability mass function of their convolutions become somewhat unwieldy.
- **Lower bounds on the expected number of messages.** Recall that the communication lower bound from [15] only applies to the maximum number of messages sent. Is it possible to prove a communication lower bound on the *expected* number of messages (even if the maximum number of messages is unbounded)? We note that the techniques from [15] does not apply.
- **More practical protocols.** In Section 5, we demonstrated that, while the parameters from previous work [15, 9] are completely impractical, our result is moderately practical. However, our pure-DP protocol still requires (expected) communication overhead of 500–1000x compared to the non-private protocol. Meanwhile, the approximate-DP protocol of [18] achieves communication overhead of only $1 + o(1)$ (assuming that δ is not too small). Due to this, there is still a large gap between the practicality of pure-DP and approximate-DP protocols. While the lower bound from [15] mentioned in the previous bullet point strongly suggests that it might not be possible to reduce the communication required for pure-DP all the way to that of approximate-DP, it remains an important

[6] This means that w.p. q we output s copies of both $+1$ and -1 messages, for both $x_i = 0$ and $x_i = 1$ cases. Without this change, the supports of the two cases are not the same and thus it obviously violates pure-DP.

question to make pure-DP protocol more practical. For example, can we reduce the communication by a factor of 10 while achieving similar utility and privacy guarantees as in this work?

- **Histogram protocol for large B.** Our protocol has communication complexity that grows linearly with B, which is impractical when B is large. Can we get protocol for histogram whose communication is $O_\epsilon\left((\log n)^{O(1)}\right)$ for $B = O(n)$ (while achieving nearly optimal errors)? For approximate-$\mathrm{DP}_{\mathrm{shuffle}}$, a histogram protocol with expected communication of $1 + O_\epsilon\left(\frac{B(\log(n/\delta)^{O(1)})}{n}\right)$ is known [18]. It would be interesting to understand if such a protocol exists in the pure-$\mathrm{DP}_{\mathrm{shuffle}}$ setting.
- **Generic $\mathbf{DP_{local}} \Rightarrow \mathbf{DP_{shuffle}}$ transformation for pure-DP?** More generally, despite the rich literature on the shuffle model, most work has focused attention on approximate-$\mathrm{DP}_{\mathrm{shuffle}}$. It would be interesting to expand the existing study to pure-$\mathrm{DP}_{\mathrm{shuffle}}$ as well. In our opinion, a main barrier in doing so is that the so-called *amplification-by-shuffling* phenomenon does not apply to pure-DP. Recall that the amplification-by-shuffling theorem [12] roughly states that, if we take any ϵ-$\mathrm{DP}_{\mathrm{local}}$ algorithm and runs it in the shuffle model, then it immediately becomes (ϵ', δ')-$\mathrm{DP}_{\mathrm{shuffle}}$ where $\epsilon' \ll \epsilon$ for any non-too-small $\delta > 0$. This means that any $\mathrm{DP}_{\mathrm{local}}$ algorithm translates to approximate-$\mathrm{DP}_{\mathrm{shuffle}}$ algorithm with improved privacy; this allows the design of approximate-$\mathrm{DP}_{\mathrm{shuffle}}$ algorithms to tap into the vast literature of $\mathrm{DP}_{\mathrm{local}}$. Unfortunately, it is known that the amplification-by-shuffling theorem does not hold when we want pure-$\mathrm{DP}_{\mathrm{shuffle}}$; see [15] for an explanation. A natural question here is thus whether we can take any ϵ-$\mathrm{DP}_{\mathrm{local}}$ algorithm, modify it slightly (while preserving utility) and make it ϵ'-$\mathrm{DP}_{\mathrm{shuffle}}$ algorithm for $\epsilon' \ll \epsilon$. Such a transformation would enable a sort of "amplification-by-shuffling" in the pure-$\mathrm{DP}_{\mathrm{shuffle}}$ regime as well.

References

1. Victor Balcer and Albert Cheu. Separating local & shuffled differential privacy via histograms. In *ITC*, pages 1:1–1:14, 2020.
2. Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In *CRYPTO*, pages 638–667, 2019.
3. Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Private summation in the multi-message shuffle model. In *CCS*, pages 657–676, 2020.
4. Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In *CRYPTO*, pages 451–468, 2008.
5. Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *SOSP*, pages 441–459, 2017.
6. Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. *TALG*, 15(4):1–40, 2019.
7. T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In *ESA*, pages 277–288, 2012.
8. Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *EUROCRYPT*, pages 375–403, 2019.
9. Albert Cheu and Chao Yan. Pure differential privacy from secure intermediaries. In *TPDP*, 2022.
10. Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

11 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

12 Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *SODA*, pages 2468–2479, 2019.

13 Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, pages 211–222, 2003.

14 Quan Geng and Pramod Viswanath. The optimal noise-adding mechanism in differential privacy. *IEEE Trans. Inf. Theory*, 62(2):925–951, 2016.

15 Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Pure differentially private summation from anonymous messages. In *ITC*, pages 15:1–15:23, 2020.

16 Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. On the power of multiple anonymous messages: Frequency estimation and selection in the shuffle model of differential privacy. In *EUROCRYPT*, pages 463–488, 2021.

17 Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. User-level differentially private learning via correlated sampling. In *NeurIPS*, pages 20172–20184, 2021.

18 Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Rasmus Pagh. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *ICML*, pages 3505–3514, 2020.

19 Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Amer Sinha. Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message. In *ICML*, pages 3692–3701, 2021.

20 Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Private aggregation from fewer anonymous messages. In *EUROCRYPT*, pages 798–827, 2020.

21 Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *STOC*, pages 351–360, 2009.

22 Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Rashkodnikova, and Adam Smith. What can we learn privately? In *FOCS*, pages 531–540, 2008.

23 Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *Journal of Privacy and Confidentiality*, 7(2):3–22, 2016.