Differentially Private Aggregation in the Shuffle Model: Almost Central Accuracy in Almost a Single Message

Badih Ghazi [1] **Ravi Kumar** [1] **Pasin Manurangsi** [1] **Rasmus Pagh** [2 1] **Amer Sinha** [3]

Abstract

The shuffle model of differential privacy has attracted attention in the literature due to it being a middle ground between the well-studied central and local models. In this work, we study the problem of summing (aggregating) real numbers or integers, a basic primitive in numerous machine learning tasks, in the shuffle model. We give a protocol achieving error arbitrarily close to that of the (Discrete) Laplace mechanism in central differential privacy, while each user only sends $1 + o(1)$ short messages in expectation.

1. Introduction

A principal goal within trustworthy machine learning is the design of privacy-preserving algorithms. In recent years, differential privacy (DP) (Dwork et al., 2006b;a) has gained significant popularity as a privacy notion due to the strong protections that it ensures. This has led to several practical deployments including by Google (Erlingsson et al., 2014; Shankland, 2014), Apple (Greenberg, 2016; Apple Differential Privacy Team, 2017), Microsoft (Ding et al., 2017), and the U.S. Census Bureau (Abowd, 2018). DP properties are often expressed in terms of parameters ε and δ, with small values indicating that the algorithm is less likely to leak information about any individual within a set of n people providing data. It is common to set ε to a small positive constant (e.g., 1), and δ to inverse-polynomial in n.

DP can be enforced for any statistical or machine learning task, and it is particularly well-studied for the *real summation* problem, where each user i holds a real number $x_i \in [0, 1]$, and the goal is to estimate $\sum_i x_i$. This constitutes a basic building block within machine learning, with extensions including (private) distributed mean estimation (see, e.g., Biswas et al., 2020; Girgis et al., 2021), stochastic gradient descent (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016; Agarwal et al., 2018), and clustering (Stemmer & Kaplan, 2018; Stemmer, 2020).

The real summation problem, which is the focus of this work, has been well-studied in several models of DP. In the *central* model where a curator has access to the raw data and is required to produce a private data release, the smallest possible absolute error is known to be $O(1/\varepsilon)$; this can be achieved via the ubiquitous Laplace mechanism (Dwork et al., 2006b), which is also known to be nearly optimal[1] for the most interesting regime of $\varepsilon \leq 1$. In contrast, for the more privacy-stringent *local* setting (Kasiviswanathan et al., 2008) (also Warner, 1965) where each message sent by a user is supposed to be private, the smallest error is known to be $\Theta_\varepsilon(\sqrt{n})$ (Beimel et al., 2008; Chan et al., 2012). This significant gap between the achievable central and local utilities has motivated the study of intermediate models of DP. The *shuffle model* (Bittau et al., 2017; Erlingsson et al., 2019; Cheu et al., 2019) reflects the setting where the user reports are randomly permuted before being passed to the analyzer; the output of the shuffler is required to be private. Two variants of the shuffle model have been studied: in the *multi-message* case (e.g., Cheu et al., 2019), each user can send multiple messages to the shuffler; in the *single-message* setting each user sends one message (e.g., Erlingsson et al., 2019; Balle et al., 2019).

For the real summation problem, it is known that the smallest possible absolute error in the single-message shuffle model[2] is $\tilde{\Theta}_\varepsilon(n^{1/6})$ (Balle et al., 2019). In contrast, multi-message shuffle protocols exist with a near-central accuracy of $O(1/\varepsilon)$ (Ghazi et al., 2020c; Balle et al., 2020), but they suffer several drawbacks in that the number of messages sent per user is required to be at least 3, each message has to be substantially longer than in the non-private case, and in particular, the number of bits of communication per user has to grow with $\log(1/\delta)/\log n$. This (at least) 3-fold communication blow-up relative to a non-private setting

*Equal contribution [1]Google Research, Mountain View [2]University of Copenhagen, Denmark [3]Google, San Bruno. Correspondence to: Badih Ghazi <badihghazi@gmail.com>, Ravi Kumar <ravi.k53@gmail.com>, Pasin Manurangsi <pasin@google.com>, Rasmus Pagh <pagh@di.ku.dk>, Amer Sinha <amersinha@google.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]Please see the supplementary material for more discussion.

[2]Here $\tilde{\Theta}_\varepsilon(\cdot)$ hides a polylogarithmic factor in $1/\delta$, in addition to a dependency on ε.

can be a limitation in real-time reporting use cases (where encryption of each message may be required and the associated cost can become dominant) and in federated learning settings (where great effort is undertaken to compress the gradients). Our work shows that near-central accuracy *and* near-zero communication overhead are possible for real aggregation over sufficiently many users:

Theorem 1. *For any $0 < \varepsilon \leq O(1), \zeta, \delta \in (0, 1/2)$, there is an (ε, δ)-DP real summation protocol in the shuffle model whose mean squared error (MSE) is at most the MSE of the Laplace mechanism with parameter $(1-\zeta)\varepsilon$, each user sends $1 + \tilde{O}_{\zeta,\varepsilon}\left(\frac{\log(1/\delta)}{\sqrt{n}}\right)$ messages in expectation, and each message contains $\frac{1}{2}\log n + O(\log \frac{1}{\zeta})$ bits.*

Note that $\tilde{O}_{\zeta,\varepsilon}$ hides a small $\mathrm{poly}(\log n, 1/\varepsilon, 1/\zeta)$ term. Moreover, the number of bits per message is equal, up to lower order terms, to that needed to achieve MSE $O(1)$ even *without* any privacy constraints.

Theorem 1 follows from an analogous result for the case of *integer* aggregation, where each user is given an element in the set $\{0, 1, ..., \Delta\}$ (with Δ an integer), and the goal of the analyzer is to estimate the sum of the users' inputs. We refer to this task as the Δ-*summation* problem.

what is the geometric mechanism

For Δ-summation, the standard mechanism in the central model is the Discrete Laplace (aka Geometric) mechanism, which first computes the true answer and then adds to it a noise term sampled from the Discrete Laplace distribution[3] with parameter ε/Δ (Ghosh et al., 2012). We can achieve an error arbitrarily close to this mechanism in the shuffle model, with minimal communication overhead:

Theorem 2. *For any $0 < \varepsilon \leq O(1), \gamma, \delta \in (0, 1/2), \Delta \in \mathbb{N}$, there is an (ε, δ)-DP Δ-summation protocol in the shuffle model whose MSE is at most that of the Discrete Laplace mechanism with parameter $(1-\gamma)\varepsilon/\Delta$, and where each user sends $1 + \tilde{O}\left(\frac{\Delta \log(1/\delta)}{\gamma\varepsilon n}\right)$ messages in expectation, with each message containing $\lceil \log \Delta \rceil + 1$ bits.*

In Theorem 2, the $\tilde{O}(\cdot)$ hides a $\mathrm{poly}\log\Delta$ factor. We also note that the number of bits per message in the protocol is within a single bit from the minimum message length needed to compute the sum without any privacy constraints. Incidentally, for $\Delta = 1$, Theorem 2 improves the communication overhead obtained by Ghazi et al. (2020b) from $O\left(\frac{\log^2(1/\delta)}{\varepsilon^2 n}\right)$ to $O\left(\frac{\log(1/\delta)}{\varepsilon n}\right)$. (This improvement turns out to be crucial in practice, as our experiments show.)

Using Theorem 1 as a black-box, we obtain the following corollary for the 1-*sparse vector summation* problem, where each user is given a 1-sparse (possibly high-dimensional) vector of norm at most 1, and the goal is to compute the sum of all user vectors with minimal ℓ_2 error.

Corollary 3. *For every $d \in \mathbb{N}$, and $0 < \varepsilon \leq O(1), \zeta, \delta \in (0, 1/2)$, there is an (ε, δ)-DP algorithm for 1-sparse vector summation in d dimensions in the shuffle model whose ℓ_2 error is at most that of the Laplace mechanism with parameter $(1-\zeta)\varepsilon/2$, and where each user sends $1 + \tilde{O}_{\zeta,\varepsilon}\left(\frac{d\log(1/\delta)}{\sqrt{n}}\right)$ messages in expectation, and each message contains $\log d + \frac{1}{2}\log n + O(\log\frac{1}{\zeta})$ bits.*

1.1. Technical Overview

We will now describe the high-level technical ideas underlying our protocol and its analysis. Since the real summation protocol can be obtained from the Δ-summation protocol using known randomized discretization techniques (e.g., from Balle et al. (2020)), we focus only on the latter. For simplicity of presentation, we will sometimes be informal here; everything will be formalized later.

Infinite Divisibility. To achieve a similar performance to the central-DP Discrete Laplace mechanism (described before Theorem 2) in the shuffle model, we face several obstacles. To begin with, the noise has to be divided among all users, instead of being added centrally. Fortunately, this can be solved through the *infinite divisibility* of Discrete Laplace distributions[4]: there is a distribution $\mathcal{D}'$ for which, if each user i samples a noise z_i independently from $\mathcal{D}'$, then $z_1 + \cdots + z_n$ has the same distribution as $\mathrm{DLap}(\varepsilon/\Delta)$.

To implement the above idea in the shuffle model, each user has to be able to send their noise z_i to the shuffler. Following Ghazi et al. (2020b), we can send such a noise in unary[5], i.e., if $z_i > 0$ we send the $+1$ message z_i times and otherwise we send the -1 message $-z_i$ times. This is in addition to user i sending their own input x_i (in binary[6], as a single message) if it is non-zero. The analyzer is simple: sum up all the messages.

Don't take their word for it. Thin about what they mean

Unfortunately, this zero-sum noise approach is *not* shuffle DP for $\Delta > 1$ because, even after shuffling, the analyzer can still see u_j, the number of messages j, which is exactly the number of users whose input is equal to j for $j \in \{2, \ldots, \Delta\}$.

Zero-Sum Noise over Non-Binary Alphabets. To overcome this issue, we have to 'noise" the values u_j themselves, while at the same time preserving the accuracy. We

[3] The Discrete Laplace distribution with parameter s, denoted by $\mathrm{DLap}(s)$, has probability mass $\frac{1-e^{-s}}{1+e^{-s}} \cdot e^{-s|k|}$ at each $k \in \mathbb{Z}$.

[4] See (Goryczka & Xiong, 2017) for a discussion on distributed noise generation via infinite divisibility.

[5] Since the distribution $\mathcal{D}'$ has a small tail probability, z_i will mostly be in $\{0, -1, +1\}$, meaning that non-unary encoding of the noise does not significantly reduce the communication.

[6] If we were to send x_i in unary similar to the noise, it would require possibly as many as Δ messages, which is undesirable for us since we later pick Δ to be $O_{\varepsilon,\delta}(\sqrt{n})$ for real summation.

achieve this by making some users send additional messages whose sum is equal to zero; e.g., a user may send $-1, -1, +2$ in conjunction with previously described messages. Since the analyzer just sums up all the messages, this additional zero-sum noise still does not affect accuracy.

The bulk of our technical work is in the privacy proof of such a protocol. To understand the challenge, notice that the analyzer still sees the u_j's, which are now highly *correlated* due to the zero-sum noise added. This is unlike most DP algorithms in the literature where noise terms are added independently to each coordinate. Our main technical insight is that, by a careful change of basis, we can "reduce" the view to the independent-noise case.

To illustrate our technique, let us consider the case where $\Delta = 2$. In this case, there are two zero-sum "noise atoms" that a user might send: $(-1, +1)$ and $(-1, -1, +2)$. These two kinds of noise are sent independently, i.e., whether the user sends $(-1, +1)$ does not affect whether $(-1, -1, +2)$ is also sent. After shuffling, the analyzer sees (u_{-1}, u_{+1}, u_{+2}). Observe that there is a one-to-one mapping between this and (v_1, v_2, v_3) defined by $v_1 := u_{-1} - 2 \cdot u_{+2}, v_2 := u_{-1} - u_{+1} - u_{+2}, v_3 := -u_{-1} + u_{+1} + 2 \cdot u_{+2}$, meaning that we may prove the privacy of the latter instead. Consider the effect of sending the $(+1, -1)$ noise: v_1 is increased by one, whereas v_2, v_3 are completely unaffected. Similarly, when we send $(-1, -1, +2)$ noise, v_2 is increased by one, whereas v_1, v_3 are completely unaffected. Hence, the noise added to v_1, v_2 are now independent! Finally, v_3 is exactly the sum of all messages, which was noised by the DLap noise explained earlier.

A vital detail omitted in the previous discussion is that the DLap noise, which affects u_{-1}, u_{+1}, is *not* canceled out in v_1, v_2. Indeed, in our formal proof we need a special argument (Lemma 9) to deal with this noise.

Moreover, generalizing this approach to larger values of Δ requires overcoming additional challenges: (i) the basis change has to be carried out over the *integers*, which precludes a direct use of classic tools from linear algebra such as the Gram–Schmidt process, and (ii) special care has to be taken when selecting the new basis so as to ensure that the sensitivity does not significantly increase, which would require more added noise (this complication leads to the definition of the $\mathbf{Q}$-*linear query* problem in Section 4).

1.2. Related Work

Summation in the Shuffle Model. Our work is most closely related to that of Ghazi et al. (2020b) who gave a protocol for the case where $\Delta = 1$ (i.e., binary summation) and our protocol can be viewed as a generalization of theirs. As explained above, this requires significant novel technical and conceptual ideas; for example, the basis change was not (directly) required by Ghazi et al. (2020b).

The idea of splitting the input into multiple additive shares dates back to the "split-and-mix" protocol of Ishai et al. (2006) whose analysis was improved in Ghazi et al. (2020c); Balle et al. (2020) to get the aforementioned shuffle DP algorithms for aggregation. These analyses all crucially rely on the addition being over a finite group. Since we actually want to sum over integers and there are n users, this approach requires the group size to be at least $n\Delta$ to prevent an "overflow". This also means that each user needs to send at least $\log(n\Delta)$ bits. On the other hand, by dealing with integers directly, each of our messages is only $\lceil \log \Delta \rceil + 1$ bits, further reducing the communication.

From a technical standpoint, our approach is also different from that of Ishai et al. (2006) as we analyze the privacy of the protocol, instead of its security as in their paper. This allows us to overcome the known lower bound of $\Omega(\frac{\log(1/\delta)}{\log n})$ on the number of messages for information-theoretic security (Ghazi et al., 2020c), and obtain a DP protocol with $\tilde{O}_{\zeta,\varepsilon}\left(\frac{\log(1/\delta)}{\sqrt{n}}\right)$ messages (where ζ is as in Theorem 1).

The Shuffle DP Model. Recent research on the shuffle model of DP includes work on aggregation mentioned above (Balle et al., 2019; Ghazi et al., 2020c; Balle et al., 2020), analytics tasks including computing histograms and heavy hitters (Ghazi et al., 2021a; Balcer & Cheu, 2020; Ghazi et al., 2020a;b; Cheu & Zhilyaev, 2021), counting distinct elements (Balcer et al., 2021; Chen et al., 2021) and private mean estimation (Girgis et al., 2021), as well as k-means clustering (Chang et al., 2021).

Aggregation in Machine Learning. We note that communication-efficient private aggregation is a core primitive in *federated learning* (see Section 4 of Kairouz et al. (2019) and the references therein). It is also naturally related to mean estimation in distributed models of DP (e.g., Gaboardi et al., 2019). Finally, we point out that communication efficiency is a common requirement in distributed learning and optimization, and substantial effort is spent on compression of the messages sent by users, through multiple methods including hashing, pruning, and quantization (see, e.g., Zhang et al., 2013; Alistarh et al., 2017; Suresh et al., 2017; Acharya et al., 2019; Chen et al., 2020).

1.3. Organization

We start with some background in Section 2. Our protocol is presented in Section 3. Its privacy property is established and the parameters are set in Section 4. Experimental results are given in Section 5. We discuss some interesting future directions in Section 6. All missing proofs can be found in the Supplementary Material (SM).

2. Preliminaries and Notation

We use $[m]$ to denote $\{1, \ldots, m\}$.

Probability. For any distribution $\mathcal{D}$, we write $z \sim \mathcal{D}$ to denote a random variable z that is distributed as $\mathcal{D}$. For two distributions $\mathcal{D}_1, \mathcal{D}_2$, let $\mathcal{D}_1 + \mathcal{D}_2$ (resp., $\mathcal{D}_1 - \mathcal{D}_2$) denote the distribution of $z_1 + z_2$ (resp., $z_1 - z_2$) where $z_1 \sim \mathcal{D}_1, z_2 \sim \mathcal{D}_2$ are independent. For $k \in \mathbb{R}$, we use $k + \mathcal{D}$ to denote the distribution of $k + z$ where $z \sim \mathcal{D}$.

A distribution $\mathcal{D}$ over non-negative integers is said to be *infinitely divisible* if and only if, for every $n \in \mathbb{N}$, there exists a distribution $\mathcal{D}_{/n}$ such that $\mathcal{D}_{/n} + \cdots + \mathcal{D}_{/n}$ is identical to $\mathcal{D}$, where the sum is over n distributions.

The negative binomial distribution with parameters $r > 0$, $p \in [0, 1]$, denoted $\mathrm{NB}(r, p)$, has probability mass $\binom{k+r-1}{k}(1-p)^r p^k$ at all $k \in \mathbb{Z}_{\geq 0}$. $\mathrm{NB}(r, p)$ is infinitely divisible; specifically, $\mathrm{NB}(r, p)_{/n} = \mathrm{NB}(r/n, p)$.

Differential Privacy. Two input datasets $X = (x_1, \ldots, x_n)$ and $X' = (x'_1, \ldots, x'_n)$ are said to be *neighboring* if and only if they differ on at most a single user's input, i.e., $x_i = x'_i$ for all but one $i \in [n]$.

Definition 4 (Differential Privacy (DP) Dwork et al. (2006b;a))**.** *Let $\varepsilon, \delta \in \mathbb{R}_{\geq 0}$. A randomized algorithm $\mathcal{A}$ taking as input a dataset is said to be* (ε, δ)-differentially private *((ε, δ)-DP) if for any two neighboring datasets X and X', and for any subset S of outputs of $\mathcal{A}$, it holds that* $\Pr[\mathcal{A}(X) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(X') \in S] + \delta$.

Shuffle DP Model. A protocol over n inputs in the shuffle DP model (Bittau et al., 2017; Erlingsson et al., 2019; Cheu et al., 2019) consists of three procedures. A *local randomizer* takes an input x_i and outputs a set of messages. The *shuffler* takes the multisets output by the local randomizer applied to each of $x_1, \ldots, x_n$, and produces a random permutation of the messages as output. Finally, the *analyzer* takes the output of the shuffler and computes the output of the protocol. Privacy in the shuffle model is enforced on the output of the shuffler when a single input is changed.

3. Generic Protocol Description

Below we describe the protocol for Δ-summation that is private in the shuffle DP model. In our protocol, the randomizer will send messages, each of which is an integer in $\{-\Delta, \ldots, +\Delta\}$. The analyzer simply sums up all the incoming messages. The messages sent from the randomizer can be categorized into three classes:

- **Input**: each user i will send x_i if it is non-zero.
- **Central Noise**: This is the noise whose sum is equal to the Discrete Laplace noise commonly used algorithms in the central DP model. This noise is sent in "unary" as $+1$ or -1 messages.

Algorithm 1 Δ-Summation Randomizer

1: **procedure** CORRNOISERANDOMIZER$_n(x_i)$
2: **if** $x_i \neq 0$
3: Send x_i
4: Sample $z_i^{+1}, z_i^{-1} \sim \mathcal{D}^{\mathrm{central}}_{/n}$
5: Send z_i^{+1} copies of $+1$, and z_i^{-1} copies of -1
6: **for** $\mathbf{s} \in \mathcal{S}$
7: Sample $z_i^{\mathbf{s}} \sim \mathcal{D}^{\mathbf{s}}_{/n}$
8: **for** $m \in \mathbf{s}$
9: Send $z_i^{\mathbf{s}}$ copies of m

Algorithm 2 Δ-Summation Analyzer

1: **procedure** CORRNOISEANALYZER
2: $R \leftarrow$ multiset of messages received
3: **return** $\sum_{y \in R} y$

- **Zero-Sum Noise**: Finally, we "flood" the messages with noise that cancels out. This noise comes from a carefully chosen sub-collection $\mathcal{S}$ of the collection of all multisets of $\{-\Delta, \ldots, +\Delta\} \setminus \{0\}$ whose sum of elements is equal to zero (e.g., $\{-1, -1, +2\}$ may belong to $\mathcal{S}$).[7] For more details, see Theorem 12 and the paragraph succeeding it. We will refer to each $\mathbf{s} \in \mathcal{S}$ as a *noise atom*.

Algorithms 1 and 6 show the generic form of our protocol, which we refer to as the *Correlated Noise* mechanism. The protocol is specified by the following infinitely divisible distributions over $\mathbb{Z}_{\geq 0}$: the "central" noise distribution $\mathcal{D}^{\mathrm{central}}$, and for every $\mathbf{s} \in \mathcal{S}$, the "flooding" noise distribution $\mathcal{D}^{\mathbf{s}}$.

Note that since $\mathbf{s}$ is a multiset, Line 8 goes over each element the same number of times it appears in $\mathbf{s}$; e.g., if $\mathbf{s} = \{-1, -1, +2\}$, the iteration $m = -1$ is executed twice.

3.1. Error and Communication Complexity

We now state generic forms for the MSE and communication cost of the protocol:

Observation 5. *MSE is* $2\,\mathrm{Var}(\mathcal{D}^{\mathrm{central}})$.

We stress here that the distribution $\mathcal{D}^{\mathrm{central}}$ itself is *not* the Discrete Laplace distribution; we pick it so that $\mathcal{D}^{\mathrm{central}} - \mathcal{D}^{\mathrm{central}}$ is DLap. As a result, $2\,\mathrm{Var}(\mathcal{D}^{\mathrm{central}})$ is indeed equal to the variance of the Discrete Laplace noise.

Observation 6. *Each user sends at most* $1 + \frac{1}{n}\left(2\mathbb{E}[\mathcal{D}^{\mathrm{central}}] + \sum_{\mathbf{s} \in \mathcal{S}} |\mathbf{s}| \cdot \mathbb{E}[\mathcal{D}^{\mathbf{s}}]\right)$ *messages in expectation, each consisting of* $\lceil \log \Delta \rceil + 1$ *bits.*

[7] Note that while $\mathcal{S}$ may be infinite, we will later set it to be finite, resulting in an efficient protocol.

4. Parameter Selection and Privacy Proof

The focus of this section is on selecting concrete distributions to initiate the protocol and formalize its privacy guarantees, ultimately proving Theorem 2. First, in Section 4.1, we introduce additional notation and reduce our task to proving a privacy guarantee for a protocol in the *central* model. With these simplifications, we give a generic form of privacy guarantees in Section 4.2. Section 4.3 and Section 4.4 are devoted to a more concrete selection of parameters. Finally, Theorem 2 is proved in Section 4.5.

4.1. Additional Notation and Simplifications

Matrix-Vector Notation. We use boldface letters to denote vectors and matrices, and standard letters to refer to their coordinates (e.g., if $\mathbf{u}$ is a vector, then u_i refers to its ith coordinate). For convenience, we allow general index sets for vectors and matrices; e.g., for an index set $\mathcal{I}$, we write $\mathbf{u} \in \mathbb{R}^{\mathcal{I}}$ to denote the tuple $(u_i)_{i\in\mathcal{I}}$. Operations such as addition, scalar-vector/matrix multiplication or matrix-vector multiplication are defined naturally.

For $i \in \mathcal{I}$, we use $\mathbf{1}_i$ to denote the ith vector in the standard basis; that is, its i-indexed coordinate is equal to 1 and each of the other coordinates is equal to 0. Furthermore, we use $\mathbf{0}$ to denote the all-zeros vector.

Let $[-\Delta, \Delta]$ denote $\{-\Delta, \dots, +\Delta\} \setminus \{0\}$, and $\mathbf{v} \in \mathbb{Z}^{[-\Delta,\Delta]}$ denote the vector $v_i := i$. Recall that a noise atom $\mathbf{s}$ is a multiset of elements from $[-\Delta, \Delta]$. It is useful to also think of $\mathbf{s}$ as a vector in $\mathbb{Z}_{\geq 0}^{[-\Delta,\Delta]}$ where its ith entry denotes the number of times i appears in $\mathbf{s}$. We overload the notation and use $\mathbf{s}$ to both represent the multiset and its corresponding vector.

Let $\mathbf{A} \in \mathbb{Z}^{[-\Delta,\Delta]\times\mathcal{S}}$ denote the matrix whose rows are indexed by $[-\Delta, \Delta]$ and whose columns are indexed by $\mathcal{S}$ where $A_{i,\mathbf{s}} = s_i$. In other words, $\mathbf{A}$ is a concatenation of column vectors $\mathbf{s}$. Furthermore, let $[-\Delta, \Delta]_{-1}$ denote $[-\Delta, \Delta] \setminus \{1\}$, and $\tilde{\mathbf{A}} \in \mathbb{Z}^{[-\Delta,\Delta]_{-1}\times\mathcal{S}}$ denote the matrix $\mathbf{A}$ with row 1 removed.

Next, we think of each input dataset $(x_1, \dots, x_n)$ as its histogram $\mathbf{h} \in \mathbb{Z}_{\geq 0}^{[\Delta]}$ where h_j denotes the number of $i \in [n]$ such that $x_i = j$. Under this notation, two input datasets $\mathbf{h}, \mathbf{h}'$ are neighbors iff $\|\mathbf{h} - \mathbf{h}'\|_\infty \leq 1$ and $\|\mathbf{h} - \mathbf{h}'\|_1 \leq 2$. For each histogram $\mathbf{h} \in \mathbb{Z}_{\geq 0}^{[\Delta]}$, we write $\mathbf{h}^{\text{ext}} \in \mathbb{Z}_{\geq 0}^{[-\Delta,\Delta]}$ to denote the vector resulting from appending Δ zeros to the beginning of $\mathbf{h}$; more formally, for every $i \in [-\Delta, \Delta]$, we let $h_i^{\text{ext}} = h_i$ if $i > 0$ and $h_i^{\text{ext}} = 0$ if $i < 0$.

An Equivalent Central DP Algorithm. A benefit of using infinitely divisible noise distributions is that they allow us to translate our protocols to equivalent ones in the *central* model, where the total sum of the noise terms has a well-understood distribution. In particular, with the notation introduced above, Algorithm 1 corresponds to Algorithm 3 in the central model:

Algorithm 3 Central Algorithm (Matrix-Vector Notation)

procedure CORRNOISECENTRAL($\mathbf{h}$)
  Sample $z^{+1}, z^{-1} \sim \mathcal{D}^{\text{central}}$
  for $\mathbf{s} \in \mathcal{S}$
    Sample $z^{\mathbf{s}} \sim \mathcal{D}^{\mathbf{s}}$
  $\mathbf{z} \leftarrow (z^{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$
  return $\mathbf{h}^{\text{ext}} + z^{+1} \cdot \mathbf{1}_1 + z^{-1} \cdot \mathbf{1}_{-1} + \mathbf{A}\mathbf{z}$.

Observation 7. CORRNOISERANDOMIZER *is* (ε, δ)*-DP in the shuffle model if and only if* CORRNOISECENTRAL *is* (ε, δ)*-DP in the central model.*

Given Observation 7, we can focus on proving the privacy guarantee of CORRNOISECENTRAL in the central model, which will be the majority of this section.

Noise Addition Mechanisms for Matrix-Based Linear Queries. The $\mathcal{D}$*-noise addition mechanism* for Δ-summation (as defined in Section 1) works by first computing the summation and adding to it a noise random variable sampled from $\mathcal{D}$, where $\mathcal{D}$ is a distribution over integers. Note that under our vector notation above, the $\mathcal{D}$-noise addition mechanism simply outputs $\langle \mathbf{v}, \mathbf{h} \rangle + z$ where $z \sim \mathcal{D}$.

It will be helpful to consider a generalization of the Δ-summation problem, which allows $\mathbf{v}$ above to be changed to any matrix (where the noise is now also a vector).

To define such a problem formally, let $\mathcal{I}$ be any index set. Given a matrix $\mathbf{Q} \in \mathbb{Z}^{\mathcal{I}\times[\Delta]}$, the $\mathbf{Q}$*-linear query* problem is to compute, given an input histogram $\mathbf{h} \in \mathbb{Z}_{\geq 0}^{[\Delta]}$, an estimate of $\mathbf{Q}\mathbf{h} \in \mathbb{Z}^{\mathcal{I}}$. (Equivalently, one can think of each user as holding a column vector of $\mathbf{Q}$ or the all-zeros vector $\mathbf{0}$, and the goal is to compute the sum of these vectors.)

The noise addition algorithms for Δ-summation can be easily generalized to the $\mathbf{Q}$-linear query case: for a collection $\boldsymbol{\mathcal{D}} = (\mathcal{D}^i)_{i\in\mathcal{I}}$ of distributions, the $\boldsymbol{\mathcal{D}}$*-noise addition mechanism* samples[8] $\mathbf{z} \sim \boldsymbol{\mathcal{D}}$ and then outputs $\mathbf{Q}\mathbf{h} + \mathbf{z}$.

4.2. Generic Privacy Guarantee

With all the necessary notation ready, we can now state our main technical theorem, which gives a privacy guarantee in terms of a right inverse of the matrix $\tilde{\mathbf{A}}$:

Theorem 8. *Let* $\mathbf{C} \in \mathbb{Z}^{\mathcal{S}\times[-\Delta,\Delta]_{-1}}$ *denote any right inverse of* $\tilde{\mathbf{A}}$ *(i.e.,* $\tilde{\mathbf{A}}\mathbf{C} = \mathbf{I}$*) whose entries are integers. Suppose that the following holds:*

- *The* $\hat{\mathcal{D}}$*-noise addition mechanism is* $(\varepsilon_1, \delta_1)$*-DP for*

[8] Specifically, sample $z_i \sim \mathcal{D}^i$ independently for each $i \in \mathcal{I}$.

Δ-*summation.*

- *The* $(\tilde{\mathcal{D}}^{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$*-noise addition mechanism is* $(\varepsilon_2, \delta_2)$*-DP for the* $[\mathbf{0}\ \mathbf{c}_2\ \cdots\ \mathbf{c}_\Delta]$*-linear query problem, where* $\mathbf{c}_i$ *denotes the* i*th column of* $\mathbf{C}$*.*

Then, CORRNOISECENTRAL *with the following parameter selections is* $(\varepsilon^* + \varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$*-DP for* Δ*-summation:*

- $\mathcal{D}^{\text{central}} = \text{NB}(1, e^{-\varepsilon^*/\Delta})$.
- $\mathcal{D}^{\{-1,+1\}} = \tilde{\mathcal{D}}^{\{-1,+1\}} + \hat{\mathcal{D}}$.
- $\mathcal{D}^{\mathbf{s}} = \tilde{\mathcal{D}}^{\mathbf{s}}$ *for all* $\mathbf{s} \in \mathcal{S} \setminus \{\{-1,+1\}\}$.

The right inverse $\mathbf{C}$ indeed represents the "change of basis" alluded to in the introduction. It will be specified in the next subsections along with the noise distributions $\hat{\mathcal{D}}, (\tilde{\mathcal{D}}^{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$.

As one might have noticed from Theorem 8, $\{-1,+1\}$ is somewhat different that other noise atoms, as its noise distribution $\mathcal{D}^{\{-1,+1\}}$ is the sum of $\hat{\mathcal{D}}$ and $\tilde{\mathcal{D}}^{\{-1,+1\}}$. A high-level explanation for this is that our central noise is sent as $-1,+1$ messages and we would like to use the noise atom $\{-1,+1\}$ to "flood out" the correlations left in $-1,+1$ messages. A precise version of this statement is given below in Lemma 9. We remark that the first output coordinate $z^{+1} - z^{-1} + \langle \mathbf{v}, \mathbf{h}\rangle$ alone has exactly the same distribution as the $\text{DLap}(\varepsilon^*/\Delta)$-noise addition mechanism. The main challenge in this analysis is that the two coordinates are correlated through z^{-1}; indeed, this is where the $\tilde{z}^{-1,+1}$ random variable helps "flood out" the correlation.

Lemma 9. *Let* $\mathcal{M}_{\text{cor}}$ *be a mechanism that, on input histogram* $\mathbf{h} \in \mathbb{Z}_{\geq 0}^{[\Delta]}$*, works as follows:*

- *Sample* z^{+1}, z^{-1} *independently from* $\text{NB}(1, e^{-\varepsilon^*/\Delta})$.
- *Sample* $\tilde{z}^{-1,+1}$ *from* $\hat{\mathcal{D}}$.
- *Output* $\left(z^{+1} - z^{-1} + \langle \mathbf{v}, \mathbf{h}\rangle\,, z^{-1} + \tilde{z}^{-1,+1}\right)$.

If the $\hat{\mathcal{D}}$*-noise addition mechanism is* $(\varepsilon_1, \delta_1)$*-DP for* Δ*-summation, then* $\mathcal{M}_{\text{cor}}$ *is* $(\varepsilon^* + \varepsilon_1, \delta_1)$*-DP.*

Lemma 9 is a *direct improvement* over the main analysis of Ghazi et al. (2020b), whose proof (which works only for $\Delta = 1$) requires the $\hat{\mathcal{D}}$-noise addition mechanism to be $(\varepsilon_1, \delta_1)$-DP for $O(\log(1/\delta)/\varepsilon)$-summation; we remove this $\log(1/\delta)/\varepsilon$ factor. Our novel insight is that when conditioned on $z^{+1} + z^{-1} = c$, rather than conditioned on $z^{+1} + z^{-1} = c - \kappa$ for some $c, \kappa \in \mathbb{Z}$, the distributions of z^{-1} are quite similar, up to a "shift" of κ and a multiplicative factor of $e^{-\kappa\varepsilon^*/\Delta}$. This allows us to "match" the two probability masses and achieve the improvement. The full proof of Lemma 9 is deferred to SM.

Let us now show how Lemma 9 can be used to prove Theorem 8. At a high-level, we run the mechanism $\mathcal{M}_{\text{cor}}$ from Lemma 9 and the $(\tilde{\mathcal{D}}^{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$-noise addition mechanism, and argue that we can use their output to construct the output for CORRNOISECENTRAL. The intuition behind this is that the first coordinate of the output of $\mathcal{M}_{\text{cor}}$ gives the weighted sum of the desired output, and the second coordinate gives the number of -1 messages used to flood the central noise. As for the $(\tilde{\mathcal{D}}^{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$-noise addition mechanism, since $\mathbf{C}$ is a right inverse of $\tilde{\mathbf{A}}$, we can use them to reconstruct the number of messages $-\Delta, \ldots, -1, 2, \ldots, \Delta$. The number of 1 messages can then be reconstructed from the weighted sum and all the numbers of other messages. These ideas are encapsulated in the proof below.

Proof of Theorem 8. Consider $\mathcal{M}_{\text{sim}}$ defined as follows:

1. First, run $\mathcal{M}_{\text{cor}}$ from Lemma 9 on input histogram $\mathbf{h}$ to arrive at an output $(b_{\text{sum}}, b_{\{-1,+1\}})$
2. Second, run $(\tilde{\mathcal{D}}^{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$-mechanism for $[\mathbf{0}\ \mathbf{c}_2\ \cdots\ \mathbf{c}_\Delta]$-linear query on $\mathbf{h}$ to get an output $\mathbf{y} = (y_{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$.
3. Output $\mathbf{u} = (u_{-\Delta}, \ldots, u_{-1}, u_1, \ldots, u_\Delta)$ computed by letting $\mathbf{w} = \tilde{\mathbf{A}}(\mathbf{y} + b_{\{-1,1\}} \cdot \mathbf{1}_{\{-1,+1\}})$ and then
$$u_i = \begin{cases} w_i & \text{if } i \neq 1, \\ b_{\text{sum}} - \sum_{i\in[-\Delta,\Delta]_{-1}} i \cdot w_i & \text{if } i = 1. \end{cases}$$

From Lemma 9 and our assumption on $\hat{\mathcal{D}}$, $\mathcal{M}_{\text{cor}}$ is $(\varepsilon^* + \varepsilon_1, \delta_1)$-DP. By assumption that the $(\tilde{\mathcal{D}}^{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$-noise addition mechanism is $(\varepsilon_2, \delta_2)$-DP for $[\mathbf{0}\ \mathbf{c}_2\ \cdots\ \mathbf{c}_\Delta]$-linear query and by the basic composition theorem, the first two steps of $\mathcal{M}_{\text{sim}}$ are $(\varepsilon^* + \varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-DP. The last step of $\mathcal{M}_{\text{sim}}$ only uses the output from the first two steps; hence, by the post-processing property of DP, we can conclude that $\mathcal{M}_{\text{sim}}$ is indeed $(\varepsilon^* + \varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$.

Next, we claim that $\mathcal{M}_{\text{sim}}(\mathbf{h})$ has the same distribution as CORRNOISECENTRAL$(\mathbf{h})$ with the specified parameters. To see this, recall from $\mathcal{M}_{\text{cor}}$ that we have $b_{\text{sum}} = \langle \mathbf{v}, \mathbf{h}\rangle + z^{+1} - z^{-1}$, and $b_{\{-1,+1\}} = z^{-1} + \tilde{z}^{\{-1,+1\}}$, where $z^{+1}, z^{-1} \sim \text{NB}(1, e^{-\varepsilon^*/\Delta})$ and $\tilde{z}^{\{-1,+1\}} \sim \hat{\mathcal{D}}$ are independent. Furthermore, from the definition of the $(\tilde{\mathcal{D}}^{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$-noise addition mechanism, we have
$$Y = [\mathbf{0}\ \mathbf{c}_2\ \cdots\ \mathbf{c}_\Delta]\,\mathbf{h} + \mathbf{f} = \mathbf{C}\tilde{\mathbf{h}}^{\text{ext}} + \mathbf{f},$$
where $f_{\mathbf{s}} \sim \tilde{\mathcal{D}}^{\mathbf{s}}$ are independent, and $\tilde{\mathbf{h}}^{\text{ext}}$ denotes $\mathbf{h}^{\text{ext}}$ after replacing its first coordinate with zero.

Notice that $\tilde{\mathbf{A}}\mathbf{1}_{\{-1,+1\}} = \mathbf{1}_{-1}$. Using this and our assumption that $\tilde{\mathbf{A}}\mathbf{C} = \mathbf{I}$, we get
$$\begin{aligned}\mathbf{w} &= \tilde{\mathbf{A}}\left(\mathbf{C}\tilde{\mathbf{h}}^{\text{ext}} + \mathbf{f} + b_{\{-1,1\}} \cdot \mathbf{1}_{\{-1,+1\}}\right) \\ &= \tilde{\mathbf{h}}^{\text{ext}} + z^{-1} \cdot \mathbf{1}_{-1} + \tilde{\mathbf{A}}(\mathbf{f} + \tilde{z}^{\{-1,+1\}} \cdot \mathbf{1}_{\{-1,+1\}}).\end{aligned}$$
Let $\mathbf{z} = \mathbf{f} + \tilde{Z}^{\{-1,+1\}} \cdot \mathbf{1}_{\{-1,+1\}}$; we can see that each entry $z_{\mathbf{s}}$ is independently distributed as $\mathcal{D}^{\mathbf{s}}$. Finally, we have
$$\begin{aligned}u_1 &= \langle \mathbf{v}, \mathbf{h}\rangle + z^{+1} - z^{-1} - \sum_{i\in[-\Delta,\Delta]_{-1}} i \cdot w_i \\ &= \left(\sum_{i\in[\Delta]} i \cdot h_i\right) + z^{+1} - z^{-1}\end{aligned}$$

$$
\begin{aligned}
&- \sum_{i\in[-\Delta,\Delta]_{-1}} i \cdot \left(\tilde{\mathbf{h}}^{\text{ext}} + z^{-1}\cdot \mathbf{1}_{-1} + \tilde{\mathbf{A}}\mathbf{z}\right)_i \\
&= \left(\sum_{i\in[\Delta]} i\cdot h_i\right) + z^{+1} - z^{-1} - \left(\sum_{i\in\{2,\dots,\Delta\}} i\cdot h_i\right) \\
&\quad + z^{-1} - \left(\sum_{i\in[-\Delta,\Delta]_{-1}} i\cdot\left(\tilde{\mathbf{A}}\mathbf{z}\right)_i\right) \\
&= h_1 + z^{+1} - \left(\sum_{i\in[-\Delta,\Delta]_{-1}} i\cdot\left(\tilde{\mathbf{A}}\mathbf{z}\right)_i\right) \\
&= h_1 + z^{+1} - \left(\sum_{i\in[-\Delta,\Delta]_{-1}} i\cdot\left(\sum_{\mathbf{s}\in\mathcal{S}} s_i\cdot z_{\mathbf{s}}\right)\right) \\
&= h_1 + z^{+1} - \left(\sum_{\mathbf{s}\in\mathcal{S}} z_{\mathbf{s}}\cdot\left(\sum_{i\in[-\Delta,\Delta]_{-1}} i\cdot s_i\right)\right).
\end{aligned}
$$

Recall that $\sum_{i\in[-\Delta,\Delta]} i\cdot s_i = 0$ for all $\mathbf{s}\in\mathcal{S}$; equivalently, $\sum_{i\in[-\Delta,\Delta]_{-1}} i\cdot s_i = -s_1$. Thus, we have

$$u_1 = h_1 + z^{+1} + \left(\sum_{\mathbf{s}\in\mathcal{S}} z_{\mathbf{s}}\cdot s_1\right) = h_1 + z^{+1} + (\mathbf{A}\mathbf{z})_1.$$

Hence, we can conclude that $\mathbf{u} = \mathbf{h}^{\text{ext}} + z^{+1}\cdot\mathbf{1}_1 + z^{-1}\cdot\mathbf{1}_{-1} + \mathbf{A}\mathbf{z}$; this implies that $\mathcal{M}_{\text{sim}}(\mathbf{h})$ has the same distribution as the mechanism CORRNOISECENTRAL($\mathbf{h}$). □

4.3. Negative Binomial Mechanism

Having established a generic privacy guarantee of our algorithm, we now have to specify the distributions $\hat{\mathcal{D}}, \tilde{\mathcal{D}}^{\mathbf{s}}$ that satisfy the conditions in Theorem 8 while keeping the number of messages sent for the noise small. Similar to Ghazi et al. (2020b), we use the negative binomial distribution. Its privacy guarantee is summarized below.[9]

Theorem 10 (Ghazi et al. (2020b))**.** *For any $\varepsilon,\delta\in(0,1)$ and $\Delta\in\mathbb{N}$, let $p = e^{-0.2\varepsilon/\Delta}$ and $r = 3(1+\log(1/\delta))$. The $\mathrm{NB}(r,p)$-additive noise mechanism is (ε,δ)-DP for Δ-summation.*

We next extend Theorem 10 to the $\mathbf{Q}$-linear query problem. To state the formal guarantees, we say that a vector $\mathbf{x}\in\mathbb{Z}^{\mathcal{I}}$ is *dominated* by vector $\mathbf{y}\in\mathbb{N}^{\mathcal{I}}$ iff $\sum_{i\in\mathcal{I}} |x_i|/y_i \le 1$. (We use the convention $0/0 = 0$ and $a/0 = \infty$ for all $a > 0$.)

Corollary 11. *Let $\varepsilon,\delta\in(0,1)$. Suppose that every column of $\mathbf{Q}\in\mathbb{Z}^{\mathcal{I}\times[\Delta]}$ is dominated by $\mathbf{t}\in\mathbb{N}^{\mathcal{I}}$. For each $i\in\mathcal{I}$, let $p_i = e^{-0.2\varepsilon/(2\mathbf{t}_i)}$, $r_i = 3(1+\log(|\mathcal{I}|/\delta))$ and $\mathcal{D}^i = \mathrm{NB}(r_i,p_i)$. Then, $(\mathcal{D}^i)_{i\in\mathcal{I}}$-noise addition mechanism is (ε,δ)-DP for $\mathbf{Q}$-linear query.*

[9] For the exact statement we use here, please refer to Theorem 13 of Ghazi et al. (2021b), which contains a correction of calculation errors in Theorem 13 of Ghazi et al. (2020b).

4.4. Finding a Right Inverse

A final step before we can apply Theorem 8 is to specify the noise atom collection $\mathcal{S}$ and the right inverse $\mathbf{C}$ of $\tilde{\mathbf{A}}$. Below we give such a right inverse where every column is dominated by a vector $\mathbf{t}$ that is "small". This allows us to then use the negative binomial mechanism in the previous section with a "small" amount of noise. How "small" $\mathbf{t}$ is depends on the expected number of messages sent; this is governed by $\|\mathbf{t}\|_{\mathcal{S}} := \sum_{\mathbf{s}\in\mathcal{S}} \|\mathbf{s}\|_1\cdot|\mathbf{t}_{\mathbf{s}}|$. With this notation, the guarantee of our right inverse can be stated as follows. (We note that in our noise selection below, every $\mathbf{s}\in\mathcal{S}$ has at most three elements. In other words, $\|\mathbf{t}\|_{\mathcal{S}}$ and $\|\mathbf{t}\|_1$ will be within a factor of three of each other.)

Theorem 12. *There exist $\mathcal{S}$ of size $O(\Delta)$, $\mathbf{C}\in\mathbb{Z}^{\mathcal{S}\times[-\Delta,\Delta]_{-1}}$ with $\tilde{\mathbf{A}}\mathbf{C} = \mathbf{I}$ and $\mathbf{t}\in\mathbb{N}^{\mathcal{S}}$ such that $\|\mathbf{t}\|_{\mathcal{S}} \le O(\Delta\log^2\Delta)$ and every column of $\mathbf{C}$ is dominated by $\mathbf{t}$.*

The full proof of Theorem 12 is deferred to SM. The main idea is to essentially proceed via Gaussian elimination on $\tilde{\mathbf{A}}$. However, we have to be careful about our choice of orders of rows/columns to run the elimination on, as otherwise it might produce a non-integer matrix $\mathbf{C}$ or one whose columns are not "small". In our proof, we order the rows based on their absolute values, and we set $\mathcal{S}$ to be the collection of $\{-1,+1\}$ and $\{i, -\lceil i/2\rceil, -\lfloor i/2\rfloor\}$ for all $i\in\{-\Delta,\dots,-2,2,\dots,\Delta\}$. In other words, these are the noise atoms we send in our protocol.

4.5. Specific Parameter Selection: Proof of Theorem 2

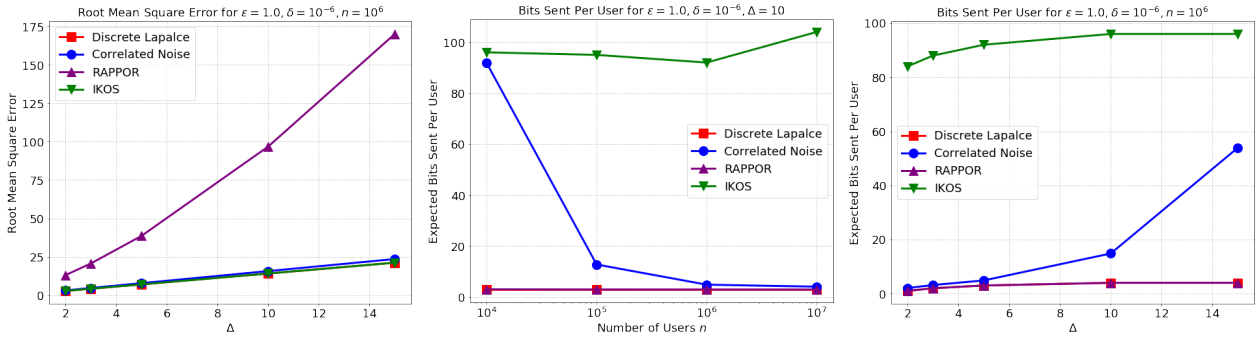
Proof of Theorem 2. Let $\mathcal{S},\mathbf{C},\mathbf{t}$ be as in Theorem 12, $\varepsilon^* = (1-\gamma)\varepsilon, \varepsilon_1 = \varepsilon_2 = \min\{1,\gamma\varepsilon\}/2, \delta_1 = \delta_2 = \delta/2$, and,

- $\mathcal{D}^{\text{central}} = \mathrm{NB}(1, e^{-\varepsilon^*/\Delta})$,
- $\hat{\mathcal{D}} = \mathrm{NB}(\hat{r},\hat{p})$ where $\hat{p} = e^{-0.2\varepsilon_1/\Delta}$ and $\hat{r} = 3(1+\log(1/\delta_1))$,
- $\tilde{\mathcal{D}}^{\mathbf{s}} = \mathrm{NB}(r_{\mathbf{s}}, p_{\mathbf{s}})$ where $p_{\mathbf{s}} = e^{-0.2\varepsilon_2/(2\mathbf{t}_{\mathbf{s}})}$ and $r^{\mathbf{s}} = 3(1+\log(|\mathcal{S}|/\delta_2))$ for all $\mathbf{s}\in\mathcal{S}$.

From Theorem 10, the $\hat{\mathcal{D}}$-noise addition mechanism is (ε_1,δ_1)-DP for Δ-summation. Corollary 11 implies that the $(\tilde{\mathcal{D}}^{\mathbf{s}})_{\mathbf{s}\in\mathcal{S}}$-noise addition mechanism is (ε_2,δ_2)-DP for $[\mathbf{0}\ \mathbf{c}_2\ \cdots\ \mathbf{c}_\Delta]$-linear query. As a result, since $\varepsilon^*+\varepsilon_1+\varepsilon_2 \le \varepsilon$ and $\delta_1+\delta_2 = \delta$, Theorem 8 ensures that CORRNOISECENTRAL with parameters as specified in the theorem is (ε,δ)-DP.

The error claim follows immediately from Observation 5 together with the fact that $\mathrm{NB}(1,e^{-\varepsilon^*/\Delta}) - \mathrm{NB}(1,e^{-\varepsilon^*}/\Delta) = \mathrm{DLap}(\varepsilon^*/\Delta)$. Using Observation 6, we can also bound the expected number of messages as

$$1 + \frac{1}{n}\left(2\mathbb{E}[\mathcal{D}^{\text{central}}] + \sum_{\mathbf{s}\in\mathcal{S}} |\mathbf{s}|\cdot\mathbb{E}[\mathcal{D}^{\mathbf{s}}]\right)$$



(a) RMSEs for varying Δ (b) Expected bits sent for varying n (c) Expected bits sent for varying Δ

Figure 1: Error and communication complexity of our "correlated noise" Δ-summation protocol compared to other protocols.

$$\leq 1 + O\left(\frac{\Delta}{\varepsilon^* n}\right) + \frac{1}{n}\left(\frac{2\hat{r}}{1-\hat{p}} + \sum_{\mathbf{s}\in\mathcal{S}} \frac{|\mathbf{s}| r_{\mathbf{s}}}{1-p_{\mathbf{s}}}\right)$$
$$= 1 + O\left(\frac{\Delta \log(1/\delta)}{\gamma \varepsilon n}\right) + O\left(\frac{\|\mathbf{t}\|_{\mathcal{S}} \cdot \log(\Delta/\delta)}{\varepsilon_2 n}\right)$$
$$\leq 1 + O\left(\frac{\Delta \log(1/\delta)}{\gamma \varepsilon n}\right) + O\left(\frac{\Delta \log^2 \Delta \cdot \log(\Delta/\delta)}{\gamma \varepsilon n}\right),$$

where each message consists of $\lceil \log \Delta \rceil + 1$ bits. □

5. Experimental Evaluation

We compare our "correlated noise" Δ-summation protocol (Algorithms 1, 6) against known algorithms in the literature, namely the IKOS "split-and-mix" protocol (Ishai et al., 2006; Balle et al., 2020; Ghazi et al., 2020c) and the fragmented[10] version of RAPPOR (Erlingsson et al., 2014; 2020). We also include the Discrete Laplace mechanism (Ghosh et al., 2012) in our plots for comparison, although it is not directly implementable in the shuffle model. We do not include the generalized (i.e., Δ-ary) Randomized Response algorithm (Warner, 1965) as it always incurs at least as large error as RAPPOR.

For our protocol, we set ε^* in Theorem 8 to 0.9ε, meaning that its MSE is that of $\mathrm{DLap}(0.9\varepsilon/\Delta)$. While the parameters set in our proofs give a theoretically vanishing overhead guarantee, they turn out to be rather impractical; instead, we resort to a tighter numerical approach to find the parameters. We discuss this, together with the setting of parameters for the other alternatives, in the SM.

For all mechanisms the root mean square error (RMSE) and the (expected) communication per user only depends on $n, \varepsilon, \delta, \Delta$, and is independent of the input data. We next summarize our findings.

Error. The IKOS algorithm has the same error as the Discrete Laplace mechanism (in the central model), whereas our algorithm's error is slightly larger due to ε^* being slightly smaller than ε. On the other hand, the RMSE for RAPPOR grows as $1/\delta$ increases, but it seems to converge as n becomes larger. (For $\delta = 10^{-6}$ and $n = 10^4, \ldots, 10^7$, we found that RMSEs differ by less than 1%.)

However, the key takeaway here is that the RMSEs of IKOS, Discrete Laplace, and our algorithm grow only linearly in Δ, but the RMSE of RAPPOR is proportional to $\Theta(\Delta^{3/2})$. This is illustrated in Figure 1a.

Communication. While the IKOS protocol achieves the same accuracy as the Discrete Laplace mechanism, it incurs a large communication overhead, as each message sent consists of $\lceil \log(n\Delta) \rceil$ bits and each user needs to send multiple messages. By contrast, when fixing Δ and taking $n \to \infty$, both RAPPOR and our algorithm only send $1 + o(1)$ messages, each of length $\lceil \log \Delta \rceil$ and $\lceil \log \Delta \rceil + 1$ respectively. This is illustrated in Figure 1b. Note that the number of bits sent for IKOS is indeed not a monotone function since, as n increases, the number of messages required decreases but the length of each message increases.

Finally, we demonstrate the effect of varying Δ in Figure 1c for a fixed value of n. Although Theorem 2 suggests that the communication overhead should grow roughly linearly with Δ, we have observed larger gaps in the experiments. This seems to stem from the fact that, while the $\tilde{O}(\Delta)$ growth would have been observed if we were using our analytic formula, our tighter parameter computation (detailed in Appendix H.1 of SM) finds a protocol with even *smaller* communication, suggesting that the actual growth might be less than $\tilde{O}(\Delta)$ though we do not know of a formal proof of this. Unfortunately, for large Δ, the optimization problem for the parameter search gets too demanding and our program does not find a good solution, leading to the "larger

[10] The RAPPOR randomizer starts with a one-hot encoding of the input (which is a Δ-bit string) and flips each bit with a certain probability. *Fragmentation* means that, instead of sending the entire Δ-bit string, the randomizer only sends the coordinates that are set to 1, each as a separate $\lceil \log \Delta \rceil$-bit message. This is known to reduce both the communication and the error in the shuffle model (Cheu et al., 2019; Erlingsson et al., 2020).

gap" observed in the experiments.

6. Conclusions

In this work, we presented a DP protocol for real and 1-sparse vector aggregation in the shuffle model with accuracy arbitrarily close to the best possible central accuracy, and with relative communication overhead tending to 0 with increasing number of users. It would be very interesting to generalize our protocol and obtain qualitatively similar guarantees for *dense* vector summation.

We also point out that in the low privacy regime ($\varepsilon \gg 1$), the *staircase* mechanism is known to significantly improve upon the Laplace mechanism (Geng & Viswanath, 2016; Geng et al., 2015); the former achieves MSE that is exponentially small in $\Theta(\varepsilon)$ while the latter has MSE $O(1/\varepsilon^2)$. An interesting open question is to achieve such a gain in the shuffle model.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *CCS*, pp. 308–318, 2016.

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

Acharya, J., Sun, Z., and Zhang, H. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In *AISTATS*, pp. 1120–1129, 2019.

Agarwal, N., Suresh, A. T., Yu, F., Kumar, S., and McMahan, H. B. cpSGD: communication-efficient and differentially-private distributed SGD. In *NeurIPS*, pp. 7575–7586, 2018.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: communication-efficient SGD via gradient quantization and encoding. In *NIPS*, pp. 1709–1720, 2017.

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.

Balcer, V., Cheu, A., Joseph, M., and Mao, J. Connecting robust shuffle privacy and pan-privacy. In *SODA*, pp. 2384–2403, 2021.

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. In *CCS*, pp. 657–676, 2020.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pp. 464–473, 2014.

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

Biswas, S., Dong, Y., Kamath, G., and Ullman, J. R. Coinpress: Practical private mean and covariance estimation. In *NeurIPS*, 2020.

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.

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

Chang, A., Ghazi, B., Kumar, R., and Manurangsi, P. Locally private k-means in one round. In *ICML*, 2021.

Chen, L., Ghazi, B., Kumar, R., and Manurangsi, P. On distributed differential privacy and counting distinct elements. In *ITCS*, 2021.

Chen, W.-N., Kairouz, P., and Özgür, A. Breaking the communication-privacy-accuracy trilemma. In *NeurIPS*, 2020.

Cheu, A. and Zhilyaev, M. Differentially private histograms in the shuffle model from fake users. *CoRR*, abs/2104.02739, 2021.

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

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.

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. *CoRR*, abs/2001.03618, 2020.

Gaboardi, M., Rogers, R., and Sheffet, O. Locally private mean estimation: z-test and tight confidence intervals. In *AISTATS*, pp. 2545–2554, 2019.

Geng, Q. and Viswanath, P. The optimal noise-adding mechanism in differential privacy. *IEEE TOIT*, 62(2): 925–951, 2016.

Geng, Q., Kairouz, P., Oh, S., and Viswanath, P. The staircase mechanism in differential privacy. *IEEE J. Sel. Top. Signal Process.*, 9(7):1176–1184, 2015.

Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., Pagh, R., and Velingker, A. Pure differentially private summation from anonymous messages. In *ITC*, 2020a.

Ghazi, B., Kumar, R., Manurangsi, P., and Pagh, R. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *ICML*, pp. 3505–3514, 2020b.

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

Ghazi, B., Golowich, N., Kumar, R., Pagh, R., and Velingker, A. On the power of multiple anonymous messages. In *EUROCRYPT*, 2021a.

Ghazi, B., Kumar, R., Manurangsi, P., and Pagh, R. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. *CoRR*, abs/2106.04247, 2021b. This version contains a correction of calculation errors in Theorem 13 of Ghazi et al. (2020b).

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

Girgis, A. M., Data, D., Diggavi, S., Kairouz, P., and Suresh, A. T. Shuffled model of federated learning: Privacy, communication and accuracy trade-offs. In *AISTATS*, 2021.

Goryczka, S. and Xiong, L. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE Trans. Dependable Secur. Comput.*, 14(5): 463–477, 2017.

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., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *CoRR*, abs/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.

Ruggles, S., Flood, S., Goeken, R., Grover, J., Meyer, E., Pacas, J., and Sobek, M. Integrated public use microdata series (IPUMS) USA: Version 10.0 [dataset]. *Minneapolis, MN*, 2020. URL `https://doi.org/10.18128/D010.V10.0`.

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

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *GlobalSIP*, pp. 245–248, 2013.

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

Stemmer, U. and Kaplan, H. Differentially private k-means with constant multiplicative error. In *NeurIPS*, pp. 5436–5446, 2018.

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

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

Zhang, Y., Duchi, J. C., Jordan, M. I., and Wainwright, M. J. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *NIPS*, pp. 2328–2336, 2013.