Distributed Differential Privacy via Shuffling

Albert Cheu[1(✉)], Adam Smith[2], Jonathan Ullman[1],
David Zeber[3], and Maxim Zhilyaev[4]

[1] Khoury College of Computer Sciences, Northeastern University
cheu.a@husky.neu.edu, jullman@ccs.neu.edu
[2] Computer Science Department, Boston University ads22@bu.edu
[3] Mozilla Foundation dzeber@mozilla.com
[4] maxim.zhilyaev@gmail.com

Abstract. We consider the problem of designing scalable, robust protocols for computing statistics about sensitive data. Specifically, we look at how best to design differentially private protocols in a distributed setting, where each user holds a private datum. The literature has mostly considered two models: the "central" model, in which a trusted server collects users' data in the clear, which allows greater accuracy; and the "local" model, in which users individually randomize their data, and need not trust the server, but accuracy is limited. Attempts to achieve the accuracy of the central model without a trusted server have so far focused on variants of cryptographic multiparty computation (MPC), which limits scalability.

In this paper, we initiate the analytic study of a *shuffled model* for distributed differentially private algorithms, which lies between the local and central models. This simple-to-implement model, a special case of the ESA framework of [5], augments the local model with an anonymous channel that randomly permutes a set of user-supplied messages. For sum queries, we show that this model provides the power of the central model while avoiding the need to trust a central server and the complexity of cryptographic secure function evaluation. More generally, we give evidence that the power of the shuffled model lies strictly between those of the central and local models: for a natural restriction of the model, we show that shuffled protocols for a widely studied *selection* problem require exponentially higher sample complexity than do central-model protocols.

1 Introduction

The past few years has seen a wave of commercially deployed systems [17, 29] for analysis of users' sensitive data in the *local model of differential privacy (LDP)*. LDP systems have several features that make them attractive in practice, and limit the barriers to adoption. Each user only sends private data to the data collector, so users do not need to fully trust the collector, and the collector is not saddled with legal or ethical obligations. Moreover, these protocols are relatively

The full version of this paper is accessible on arXiv

simple and scalable, typically requiring each party to asynchronously send just a single short message.

However, the local model imposes strong constraints on the utility of the algorithm. These constraints preclude the most useful differentially private algorithms, which require a *central model* where the users' data is sent in the clear, and the data collector is trusted to perform only differentially private computations. Compared to the central model, the local model requires enormous amounts of data, both in theory and in practice (see e.g. [20] and the discussion in [5]). Unsurprisingly, the local model has so far only been used by large corporations like Apple and Google with billions of users.

In principle, there is no dilemma between the central and local models, as any algorithm can be implemented without a trusted data collector using cryptographic *multiparty computation (MPC)*. However, despite dramatic recent progress in the area of practical MPC, existing techniques still require large costs in terms of computation, communication, and number of rounds of interaction between the users and data collector, and are considerably more difficult for companies to extend and maintain.

In this work, we initiate the analytic study of an intermediate model for distributed differential privacy called the *shuffled model*. This model, a special case of the ESA framework of [5], augments the standard model of local differential privacy with an anonymous channel (also called a shuffler) that collects messages from the users, randomly permutes them, and then forwards them to the data collector for analysis. For certain applications, this model overcomes the limitations on accuracy of local algorithms while preserving many of their desirable features. However, under natural constraints, this model is dramatically weaker than the central model. In more detail, we make two primary contributions:

- We give a simple, non-interactive algorithm in the shuffled model for estimating a single Boolean-valued statistical query (also known as a counting query) that essentially matches the error achievable by centralized algorithms. We also show how to extend this algorithm to estimate a bounded real-valued statistical query, albeit at an additional cost in communication. These protocols are sufficient to implement any algorithm in the *statistical queries model* [22], which includes methods such as gradient descent.
- We consider the ubiquitous *variable-selection problem*—a simple but canonical optimization problem. Given a set of counting queries, the variable-selection problem is to identify the query with nearly largest value (i.e. an "approximate argmax"). We prove that the sample complexity of variable selection in a natural restriction of the shuffled model is exponentially larger than in the central model. The restriction is that each user send only a single message into the shuffle, as opposed to a set of messages, which we call this the *one-message* shuffled model. Our positive results show that the sample complexity in the shuffled model is polynomially smaller than in the local model. Taken together, our results give evidence that the central, shuffled, and local models are strictly ordered in the accuracy they can achieve for selection. Our lower bounds follow from a structural result showing that any algorithm that is

private in the one-message shuffled model is also private in the local model with weak, but non-trivial, parameters.

In concurrent and independent work, Erlingsson et al. [16] give conceptually similar positive results for local protocols aided by a shuffler. We give a more detailed comparison between our work and theirs after giving a thorough description of the model and our results (Section 2.3)

1.1 Background and Related Work

Models for Differentially Private Algorithms. Differential privacy [14] is a restriction on the algorithm that processes a dataset to provide statistical summaries or other output. It ensures that, no matter what an attacker learns by interacting with the algorithm, it would have learned nearly the same thing whether or not the dataset contained any particular individual's data [21]. Differential privacy is now widely studied, and algorithms satisfying the criterion are increasingly deployed [1, 24, 17].

There are two well-studied models for implementing differentially-private algorithms. In the *central model*, raw data are collected at a central server where they are processed by a differentially private algorithm. In the *local model* [33, 18, 14], each individual applies a differentially private algorithm locally to their data and shares only the output of the algorithm—called a report or response—with a server that aggregates users' reports. The local model allows individuals to retain control of their data since privacy guarantees are enforced directly by their devices. It avoids the need for a single, widely-trusted entity and the resulting single point of security failure. The local model has witnessed an explosion of research in recent years, ranging from theoretical work to deployed implementations. A complete survey is beyond the scope of this paper.

Unfortunately, for most tasks there is a large, unavoidable gap between the accuracy that is achievable in the two models. [4] and [8] show that estimating the sum of bits, one held by each player, requires error $\Omega(\sqrt{n}/\varepsilon)$ in the local model, while an error of just $O(1/\varepsilon)$ is possible the central model. [12] extended this lower bound to a wide range of natural problems, showing that the error must blowup by at least $\Omega(\sqrt{n})$, and often by an additional factor growing with the data dimension. More abstractly, [20] showed that the power of the local model is equivalent to the *statistical query model* [22] from learning theory. They used this to show an exponential separation between the accuracy and sample complexity of local and central algorithms. Subsequently, an even more natural separation arose for the variable-selection problem [12, 31], which we also consider in this work.

Implementing Central-Model Algorithms in Distributed Models. In principle, one could also use the powerful, general tools of modern cryptography, such as multiparty computation (MPC), or secure function evaluation, to simulate central model algorithms in a setting without a trusted server [13], but such algorithms currently impose bandwidth and liveness constraints that make

them impractical for large deployments. In contrast, Google [17] now uses local differentially private protocols to collect certain usage statistics from hundreds of millions of users' devices.

A number of specific, efficient MPC algorithms have been proposed for differentially private functionalities. They generally either (1) focus on simple summations and require a single "semi-honest"/"honest-but-curious" server that aggregates user answers, as in [26, 9, 6] ; or (2) allow general computations, but require a network of servers, a majority of whom are assumed to behave honestly, as in [11]. As they currently stand, these approaches have a number of drawbacks: they either require users to trust that a server maintained by a service provided is behaving (semi-)honestly, or they require that a coalition of service providers collaborate to run protocols that reveal to each other who their users are and *what computations they are performing on their users' data*. It is possible to avoid these issues by combining anonymous communication layers and MPC protocols for universal circuits but, with current techniques, such modifications destroy the efficiency gains relative to generic MPC.

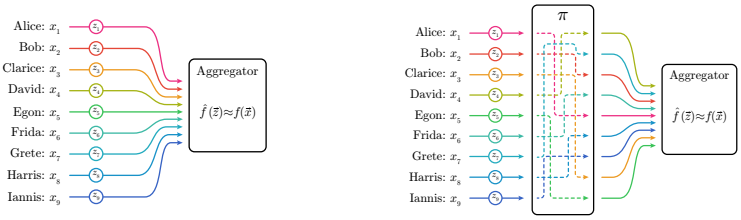
Thus, a natural question—relevant no matter how the state of the art in MPC evolves—is to identify simple (and even minimal) primitives that can be implemented via MPC in a distributed model and are expressive enough to allow for sophisticated private data analysis. In this paper, we show that shuffling is a powerful primitive for differentially private algorithms.

Mixnets. One way to realize the shuffling functionality is via a mixnet. A *mix network*, or *mixnet*, is a protocol involving several computers that takes as input a sequence of encrypted messages, and outputs a uniformly random permutation of those messages' plaintexts. Introduced by [10], the basic idea now exists in many variations. In its simplest instantiation, the network consists of a sequence of servers, whose identities and ordering are public information.[1] Messages, each one encrypted with all the servers' keys, are submitted by users to the first server. Once enough messages have been submitted, each server in turn performs a *shuffle* in which the server removes one layer of encryption and sends a permutation of the messages to the next server. In a *verifiable shuffle*, the server also produces a cryptographic proof that the shuffle preserved the multi-set of messages. The final server sends the messages to their final recipients, which might be different for each message. A variety of efficient implementations of mixnets with verifiable shuffles exist (see, e.g., [23, 5] and citations therein).

Another line of work [19, 30] shows how to use differential privacy *in addition* to mixnets to make communication patterns differentially private for the purposes of anonymous computation. Despite the superficial similarly, this line of work is orthogonal to ours, which is about how to use mixnets themselves to achieve (more accurate) differentially private data analysis.

Shufflers as a Primitive for Private Data Analysis. This paper studies how to use a shuffler (e.g. a mixnet) as a cryptographic primitive to implement

[1] Variations on this idea based on *onion routing* allow the user to specify a secret path through a network of mixes.



Prototypical (one-message) protocols in the local model (left) and the shuffled model (right).

differentially-private algorithms. Bittau et al [5] propose a general framework, dubbed *encode-shuffle-analyze* (or *ESA*), which generalizes the local and central models by allowing a local randomized encoding step E performed on user devices, a permutation step S in which encrypted encodings are shuffled, and a final randomized process A that analyzes the permuted encodings. We ask what privacy guarantee can be provided if we rely only on the local encoding E and the shuffle S—the analyst A is untrusted. In particular, we are interested in protocols that are substantially more accurate than is possible in the local model (in which the privacy guarantee relies entirely on the encoding E). This general question was left open by [5] .

One may think of the shuffled model as specifying a highly restricted MPC primitive on which we hope to base privacy. Relative to general MPC, the use of mixnets for shuffling provides several advantages: First, there already exist a number of highly efficient implementations. Second, their trust model is simple and robust—as long as a single one of the servers performs its shuffle honestly, the entire process is a uniformly random permutation, and our protocols' privacy guarantees will hold. The architecture and trust guarantees are also easy to explain to nonexperts (say, with metaphors of shuffled cards or shell games). Finally, mixnets automatically provide a number of additional features that are desirable for data collection: they can maintain secrecy of a company's user base, since each company's users could use that company's server as their first hop; and they can maintain secrecy of the company's computations, since the specific computation is done by the analyst. Note that we think of a mixnet here as operating on large batches of messages, whose size is denoted by n. (In implementation, this requires a fair amount of latency, as the collection point must receive sufficiently many messages before proceeding—see Bittau et al. [5]).

Understanding the possibilities and limitations of shuffled protocols for private data analysis is interesting from both theoretical and practical perspectives. It provides an intermediate abstraction, and we give evidence that it lies strictly between the central and local models. Thus, it sheds light on the minimal cryptographic primitives needed to get the central model's accuracy. It also

provides an attractive platform for near-term deployment [5], for the reasons listed above.

For the remainder of this paper, we treat the shuffler as an abstract service that randomly permutes a set of messages. We leave a discussion of the many engineering, social, and cryptographic implementation considerations to future work.

2 Overview of Results

The Shuffled Model. In our model, there are n users, each with data $x_i \in \mathcal{X}$. Each user applies some *encoder* $R : \mathcal{X} \to \mathcal{Y}^m$ to their data and sends the *messages* $(y_{i,1}, \ldots, y_{i,m}) = R(x_i)$. In the *one-message shuffled model*, each user sends $m = 1$ message. The $n \cdot m$ messages $y_{i,j}$ are sent to a *shuffler* $S : \mathcal{Y}^* \to \mathcal{Y}^*$ that takes these messages and outputs them in a uniformly random order. The shuffled set of messages is then passed through some *analyzer* $A : \mathcal{Y}^* \to \mathcal{Z}$ to estimate some function $f(x_1, \ldots, x_n)$. Thus, the protocol P consists of the tuple (R, S, A). We say that the algorithm is (ε, δ)-*differentially private in the shuffled model* if the algorithm $M_R(x_1, \ldots, x_n) = S(\cup_{i=1}^n R(x_i))$ satisfies (ε, δ)-differential privacy. For more detail, see the discussion leading to Definition 8.

In contrast to the local model, differential privacy is now a property of all n users' messages, and the (ε, δ) may be functions of n. However, if an adversary were to inject additional messages, then it would not degrade privacy, provided that those messages are independent of the honest users' data. Thus, we may replace n, in our results, as a *lower bound* on the number of honest users in the system. For example, if we have a protocol that is private for n users, but instead we have $\frac{n}{p}$ users of which we assume at least a p fraction are honest, the protocol will continue to satisfy differential privacy.

2.1 Algorithmic Results

Our main result shows how to estimate any bounded, real-valued linear statistic (a *statistical query*) in the shuffled model with error that nearly matches the best possible utility achievable in the central model.

Theorem 1. *For every $\varepsilon \in (0, 1)$, and every $\delta \gtrsim \varepsilon n 2^{-\varepsilon n}$ and every function $f : \mathcal{X} \to [0, 1]$, there is a protocol P in the shuffled model that is (ε, δ)-differentially private, and for every n and every $X = (x_1, \ldots, x_n) \in \mathcal{X}^n$,*

$$\mathbb{E}\left[\left|P(X) - \sum_{i=1}^{n} f(x_i)\right|\right] = O\left(\frac{1}{\varepsilon} \log \frac{n}{\delta}\right).$$

Each user sends $m = \Theta(\varepsilon\sqrt{n})$ one-bit messages.

For comparison, in the central model, the Laplace mechanism achieves $(\varepsilon, 0)$-differential privacy and error $O(\frac{1}{\varepsilon})$. In contrast, error $\Omega(\frac{1}{\varepsilon}\sqrt{n})$ is necessary in the

local model. Thus, for answering statistical queries, this protocol essentially has the best properties of the local and central models (up to logarithmic factors).

In the special case of estimating a sum of bits (or a Boolean-valued linear statistic), our protocol has a slightly nicer guarantee and form.

Theorem 2. *For every $\varepsilon \in (0,1)$, and every $\delta \gtrsim 2^{-\varepsilon n}$ and every function $f : \mathcal{X} \to \{0,1\}$, there is a protocol P in the shuffled model that is (ε, δ)-differentially private, and for every n and every $X = (x_1, \ldots, x_n) \in \mathcal{X}^n$,*

$$\mathbb{E}\left[\left|P(X) - \sum_{i=1}^{n} f(x_i)\right|\right] = O\left(\frac{1}{\varepsilon}\sqrt{\log\frac{1}{\delta}}\right).$$

Each user sends a single one-bit message.

The protocol corresponding to Theorem 2 is extremely simple:

1. For some appropriate choice of $p \in (0,1)$, each user i with input x_i outputs $y_i = x_i$ with probability $1-p$ and a uniformly random bit y_i with probability p. When ε is not too small, $p \approx \frac{\log(1/\delta)}{\varepsilon^2 n}$.
2. The analyzer collects the shuffled messages $y_1, \ldots, y_n$ and outputs

$$\frac{1}{1-p}\left(\sum_{i=1}^{n} y_i - \frac{p}{2}\right).$$

Intuition. In the local model, an adversary can map the set of observations $\{y_1, \ldots, y_n\}$ to users. Thus, to achieve ε-differential privacy, the parameter p should be set close to $\frac{1}{2}$. In our model, the attacker sees only the anonymized set of observations $\{y_1, \ldots, y_n\}$, whose distribution can be simulated using only $\sum_i y_i$. Hence, to ensure that the protocol is differentially private, it suffices to ensure that $\sum_i y_i$ is private, which we show holds for $p \approx \frac{\log(1/\delta)}{\varepsilon^2 n} \ll \frac{1}{2}$.

Communication Complexity. Our protocol for real-valued queries requires $\Theta(\varepsilon\sqrt{n})$ bits per user. In contrast, the local model requires just a single bit, but incurs error $\Omega(\frac{1}{\varepsilon}\sqrt{n})$. A generalization of Theorem 1 gives error $O(\frac{\sqrt{n}}{r} + \frac{1}{\varepsilon}\log\frac{r}{\delta})$ and sends r bits per user, but we do not know if this tradeoff is necessary. Closing this gap is an interesting open question.

2.2 Negative Results

We also prove negative results for algorithms in the *one-message* shuffled model. These results hinge on a structural characterization of private protocols in the one-message shuffled model.

Theorem 3. *If a protocol $P = (R, S, A)$ satisfies (ε, δ)-differential privacy in the one-message shuffled model, then R satisfies $(\varepsilon + \ln n, \delta)$-differential privacy. Therefore, P is $(\varepsilon + \ln n, \delta)$-differentially private in the local model.*

Using Theorem 3 (and a transformation of [7] from (ε, δ)-DP to $(O(\varepsilon), 0)$-DP in the local model), we can leverage existing lower bounds for algorithms in the local model to obtain lower bounds on algorithms in the shuffled model.

Variable Selection. In particular, consider the following *variable selection problem*: given a dataset $x \in \{0,1\}^{n \times d}$, output $\widehat{J}$ such that

$$\sum_{i=1}^{n} x_{i,\widehat{J}} \geq \left(\max_{j \in [d]} \sum_{i=1}^{n} x_{i,j} \right) - \frac{n}{10} .$$

(The $\frac{n}{10}$ approximation term is somewhat arbitrary—any sufficiently small constant fraction of n will lead to the same lower bounds and separations.)

Any local algorithm (with $\varepsilon = 1$) for selection requires $n = \Omega(d \log d)$, whereas in the central model the exponential mechanism [25] solves this problem for $n = O(\log d)$. The following lower bound shows that for this ubiquitous problem, the one-message shuffled model cannot match the central model.

Theorem 4. *If P is a $(1, \frac{1}{n^{10}})$-differentially private protocol in the one-message shuffled model that solves the selection problem (with high probability) then $n = \Omega(d^{1/17})$. Moreover this lower bound holds even if x is drawn iid from a product distribution over $\{0,1\}^d$.*

In Section 6, we also prove lower bounds for the well studied *histogram problem*, showing that any one-message shuffled-model protocol for this problem must have error growing (polylogarithmically) with the size of the data domain. In contrast, in the central model it is possible to release histograms with no dependence on the domain size, even for *infinite* domains.

We remark that our lower bound proofs do not apply if the algorithm sends multiple messages through the shuffler. However, we do not know whether beating the bounds is actually possible. Applying our bit-sum protocol d times (together with differential privacy's composition property) shows that $n = \tilde{O}(\sqrt{d})$ samples suffice in the general shuffled model. We also do not know if this bound can be improved. We leave it as an interesting direction for future work to fully characterize the power of the shuffled model.

2.3 Comparison to [16]

In concurrent and independent work, Erlingsson et al. [16] give conceptually similar positive results for local protocols aided by a shuffler. Specifically, they prove a general amplification result: adding a shuffler to any protocol satisfying local differential privacy improve the privacy parameters, often quite significantly. This amplification result can be seen as a partial converse to our transformation from shuffled protocols to local protocols (Theorem 3).

Their result applies to *any* local protocol, whereas our protocol for bit-sums (Theorem 2) applies specifically to the one-bit randomized response protocol. However, when specialized to randomized response, their result is quantitatively weaker than ours. As stated, their results only apply to local protocols that satisfy

ε-differential privacy for $\varepsilon < 1$. In contrast, the proof of Theorem 2 shows that, for randomized response, local differential privacy $\varepsilon \approx \ln(n)$ can be amplified to $\varepsilon' = 1$. Our best attempt at generalizing their proof to the case of $\varepsilon \gg 1$ does not give any amplification for local protocols with $\varepsilon \approx \ln(n)$. Specifically, our best attempt at applying their method to the case of randomized response yields a shuffled protocol that is 1-differentially private and has error $\Theta(n^{5/12})$, which is just slightly better than the error $O(\sqrt{n})$ that can be achieved without a shuffler.

3 Model and Preliminaries

In this section, we define terms and notation used throughout the paper. We use $\mathrm{Ber}(p)$ to denote the Bernoulli distribution over $\{0, 1\}$, which has value 1 with probability p and 0 with probability $1 - p$. We will use $\mathrm{Bin}(n, p)$ to denote the binomial distribution (i.e. the sum of n independent samples from $\mathrm{Ber}(p)$.

3.1 Differential Privacy

Let $X \in \mathcal{X}^n$ be a *dataset* consisting of elements from some universe $\mathcal{X}$. We say two datasets X, X' are *neighboring* if they differ on at most one user's data, and denote this $X \sim X'$.

Definition 5 (Differential Privacy [14]). *An algorithm $M : \mathcal{X}^* \to \mathcal{Z}$ is (ε, δ)-differentially private if for every $X \sim X' \in \mathcal{X}^*$ and every $T \subseteq \mathcal{Z}$*

$$\mathbb{P}\left[M(X) \in T\right] \leq e^{\varepsilon}\mathbb{P}\left[M(X') \in T\right] + \delta.$$

where the probability is taken over the randomness of M.

Differential privacy satisfies two extremely useful properties:

Lemma 6 (Post-Processing [14]). *If M is (ε, δ)-differentially private, then for every A, $A \circ M$ is (ε, δ)-differentially private.*

Lemma 7 (Composition [14, 15]). *If $M_1, \ldots, M_T$ are (ε, δ)-differentially private, then the composed algorithm*

$$\widetilde{M}(X) = (M_1(X), \ldots, M_T(X))$$

is $(\varepsilon', \delta' + T\delta)$-differentially private for every $\delta' > 0$ and $\varepsilon' = \varepsilon(e^{\varepsilon} - 1)T + \varepsilon\sqrt{2T\log(1/\delta')}$.

3.2 Differential Privacy in the Shuffled Model

In our model, there are n *users*, each of whom holds data $x_i \in \mathcal{X}$. We will use $X = (x_1, \ldots, x_n) \in \mathcal{X}^n$ to denote the *dataset* of all n users' data. We say two datasets X, X' are *neighboring* if they differ on at most one user's data, and denote this $X \sim X'$.

The protocols we consider consist of three algorithms:

- $R : \mathcal{X} \to \mathcal{Y}^m$ is a randomized *encoder* that takes as input a single users' data x_i and outputs a set of m *messages* $y_{i,1}, \ldots, y_{i,m} \in \mathcal{Y}$. If $m = 1$, then P is in the *one-message shuffled model*.
- $S : \mathcal{Y}^* \to \mathcal{Y}^*$ is a *shuffler* that takes a set of messages and outputs these messages in a uniformly random order. Specifically, on input $y_1, \ldots, y_N$, S chooses a uniformly random permutation $\pi : [N] \to [N]$ and outputs $y_{\pi(1)}, \ldots, y_{\pi(N)}$.
- $A : \mathcal{Y}^* \to \mathcal{Z}$ is some *analysis function* or *analyzer* that takes a set of messages $y_1, \ldots, y_N$ and attempts to estimate some function $f(x_1, \ldots, x_n)$ from these messages.

We denote the overall protocol $P = (R, S, A)$. The mechanism by which we achieve privacy is

$$\Pi_R(x_1, \ldots, x_n) = S(\cup_{i=1}^n R(x_i)) = S(y_{1,1}, \ldots, y_{n,m}),$$

where both R and S are randomized. We will use $P(X) = A \circ \Pi_R(X)$ to denote the output of the protocol. However, by the post-processing property of differential privacy (Lemma 6), it will suffice to consider the privacy of $\Pi_R(X)$, which will imply the privacy of $P(X)$. We are now ready to define differential privacy for protocols in the shuffled model.

Definition 8 (Differential Privacy in the Shuffled Model). *A protocol $P = (R, S, A)$ is (ε, δ)-differentially private if the algorithm $\Pi_R(x_1, \ldots, x_n) = S(R(x_1), \ldots, R(x_n))$ is (ε, δ)-differentially private (Definition 5).*

In this model, privacy is a property of the entire set of users' messages and of the shuffler, and thus ε, δ may depend on the number of users n. When we wish to refer to P or Π with a specific number of users n, we will denote this by P_n or Π_n.

We remark that if an adversary were to inject additional messages, then it would not degrade privacy, provided that those messages are independent of the honest users' data. Thus, we may replace n, in our results, with an assumed *lower bound* on the number of honest users in the system.

In some of our results it will be useful to have a generic notion of accuracy for a protocol P.

Definition 9 (Accuracy of Distributed Protocols). *Protocol $P = (R, S, A)$ is* (α, β)-accurate *for the function $f : \mathcal{X}^* \to \mathcal{Z}$ if, for every $X \in \mathcal{X}^*$, we have $\mathbb{P}\left[d(P(X), f(X)) \leq \alpha\right] \geq 1 - \beta$ where $d : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is some application-dependent distance measure.*

As with the privacy guarantees, the accuracy of the protocol may depend on the number of users n, and we will use P_n when we want to refer to the protocol with a specific number of users.

Composition of Differential Privacy We will use the following useful composition property for protocols in the shuffled model, which is an immediate

consequence of Lemma 7 and the post-processing Lemma 6. This lemma allows us to directly compose protocols in the shuffled model while only using the shuffler once, rather than using the shuffler independently for each protocol being composed.

Lemma 10 (Composition of Protocols in the Shuffled Model). *If $\Pi_1 = (R_1, S), \ldots, \Pi_T = (R_T, S)$ for $R_t : \mathcal{X} \to \mathcal{Y}^m$ are each (ε, δ)-differentially private in the shuffled model, and $\widetilde{R} : \mathcal{X} \to \mathcal{Y}^{mT}$ is defined as*

$$\widetilde{R}(x_i) = (R_1(x_i), \ldots, R_T(x_i))$$

then, for every $\delta' > 0$, the composed protocol $\widetilde{\Pi} = (\widetilde{R}, S)$ is $(\varepsilon', \delta' + T\delta)$-differentially private in the shuffled model for $\varepsilon' = \varepsilon^2 + 2\varepsilon\sqrt{T \log(1/\delta')}$.

Local Differential Privacy If the shuffler S were replaced with the identity function (i.e. if it did not randomly permute the messages) then we would be left with exactly the *local model of differential privacy*. That is, a locally differentially private protocol is a pair of algorithms $P = (R, A)$, and the output of the protocol is $P(X) = A(R(x_1), \ldots, R(x_n))$. A protocol P is differentially private in the local model if and only if the algorithm R is differentially private. In Section 6 we will see that if $P = (R, S, A)$ is a differentially private protocol in the one-message shuffled model, then R itself must satisfy local differential privacy for non-trivial (ε, δ), and thus $(R, A \circ S)$ is a differentially private local protocol for the same problem.

4 A Protocol for Boolean Sums

In this section we describe and analyze a protocol for computing a sum of $\{0, 1\}$ bits, establishing Theorem 2 in the introduction.

4.1 The Protocol

In our model, the data domain is $\mathcal{X} = \{0, 1\}$ and the function being computed is $f(x_1, \ldots, x_n) = \sum_{i=1}^n x_i$. Our protocol, P_λ, is specified by a parameter $\lambda \in [0, n]$ that allows us to trade off the level of privacy and accuracy. Note that λ may be a function of the number of users n. We will discuss in Section 4.3 how to set this parameter to achieve a desired level of privacy. For intuition, one may wish to think of the parameter $\lambda \approx \frac{1}{\varepsilon^2}$ when ε is not too small.

The basic outline of P_λ is as follows. Roughly, a random set of λ users will choose y_i randomly, and the remaining $n - \lambda$ will choose y_i to be their input bit x_i. The output of each user is the single message y_i. The outputs are then shuffled and the output of the protocol is the sum $\sum_{i=1}^n y_i$, shifted and scaled so that it is an unbiased estimator of $\sum_{i=1}^n x_i$.

The protocol is described in Algorithm 1. The full name of this protocol is $P_\lambda^{0/1}$, where the superscript serves to distinguish it with the real sum protocol $P_{\lambda,r}^{\mathbb{R}}$

(Section 5). Because of the clear context of this section, we drop the superscript. Since the analysis of both the accuracy and utility of the algorithm will depend on the number of users n, we will use $P_{n,\lambda}, R_{n,\lambda}, A_{n,\lambda}$ to denote the protocol and its components in the case where the number of users is n.

Algorithm 1: A shuffled protocol $P^{0/1}_{n,\lambda} = (R^{0/1}_{n,\lambda}, S, A^{0/1}_{n,\lambda})$ for computing the sum of bits

```
// Local Randomizer
```
$R^{0/1}_{n,\lambda}(x)$**:**
- **Input:** $x \in \{0,1\}$, parameters $n \in \mathbb{N}, \lambda \in (0,n)$.
- **Output:** $\mathbf{y} \in \{0,1\}$
- Let $\mathbf{b} \leftarrow Ber(\frac{\lambda}{n})$
- **If** $\mathbf{b} = 0$ **: Return** $\mathbf{y} \leftarrow x$;
- **ElseIf** $\mathbf{b} = 1$ **: Return** $\mathbf{y} \leftarrow Ber\left(\frac{1}{2}\right)$;

```
// Analyzer
```
$A^{0/1}_{n,\lambda}(y_1, \ldots, y_n)$**:**
- **Input:** $(y_1, \ldots, y_n) \in \{0,1\}^n$, parameters $n \in \mathbb{N}, \lambda \in (0,n)$.
- **Output:** $z \in [0,n]$
- **Return** $z \leftarrow \frac{n}{n-\lambda} \cdot \left(\sum_{i=1}^{n} y_i - \frac{\lambda}{2}\right)$

4.2 Privacy Analysis

In this section we will prove that P_λ satisfies (ε, δ)-differential privacy. Note that if $\lambda = n$ then the each user's output is independent of their input, so the protocol trivially satisfies $(0,0)$-differential privacy, and thus our goal is to prove an upper bound on the parameter λ that suffices to achieve a given (ε, δ).

Theorem 11 (Privacy of P_λ). *There are absolute constants $\kappa_1, \ldots, \kappa_5$ such that the following holds for P_λ. For every $n \in \mathbb{N}$, $\delta \in (0,1)$ and $\frac{\kappa_2 \log(1/\delta)}{n} \leq \varepsilon \leq 1$, there exists a $\lambda = \lambda(n, \varepsilon, \delta)$ such that $P_{n,\lambda}$ is (ε, δ) differentially private and,*

$$\lambda \leq \begin{cases} \frac{\kappa_4 \log(1/\delta)}{\varepsilon^2} & \text{if } \varepsilon \geq \sqrt{\frac{\kappa_3 \log(1/\delta)}{n}} \\ n - \frac{\kappa_5 \varepsilon n^{3/2}}{\sqrt{\log(1/\delta)}} & \text{otherwise} \end{cases}$$

In the remainder of this section we will prove Theorem 11.

The first step in the proof is the observation that the output of the shuffler depends only on $\sum_i y_i$. It will be more convenient to analyze the algorithm C_λ (Algorithm 2) that simulates $S(R_\lambda(x_1), \ldots, R_\lambda(x_n))$. Claim 12 shows that the output distribution of C_λ is indeed the same as that of the output $\sum_i y_i$. Therefore, privacy of C_λ carries over to P_λ.

Algorithm 2: $C_\lambda(x_1 \dots x_n)$

Input: $(x_1 \dots x_n) \in \{0,1\}^n$, parameter $\lambda \in (0, n)$.
Output: $\mathbf{y} \in \{0, 1, 2, \dots, n\}$

Sample $\mathbf{s} \leftarrow Bin\left(n, \frac{\lambda}{n}\right)$
Define $\mathcal{H}_s = \{H \subseteq [n] : |H| = s\}$ and choose $\mathbf{H} \leftarrow \mathcal{H}_s$ uniformly at random
Return $\mathbf{y} \leftarrow \sum_{i \notin \mathbf{H}} x_i + Bin\left(s, \frac{1}{2}\right)$

Claim 12. *For every $n \in \mathbb{N}$, $x \in \{0,1\}^n$, and every $r \in \{0, 1, 2, \dots, n\}$,*

$$\mathbb{P}\left[C_\lambda(X) = r\right] = \mathbb{P}\left[\sum_{i=1}^{n} R_{n,\lambda}(x_i) = r\right]$$

Proof. Fix any $r \in \{0, 1, 2, \dots, n\}$.

$$\begin{aligned}
\mathbb{P}\left[C_\lambda(X) = r\right] &= \sum_{H \subseteq [n]} \mathbb{P}\left[C_\lambda(X) = r \cap \mathbf{H} = H\right] \\
&= \sum_{H \subseteq [n]} \mathbb{P}\left[\sum_{i \notin H} x_i + Bin\left(|H|, \frac{1}{2}\right) = r\right] \cdot \left(\frac{\lambda}{n}\right)^{|H|} \left(1 - \frac{\lambda}{n}\right)^{n-|H|} \\
&= \sum_{H \subseteq [n]} \mathbb{P}\left[\sum_{i \notin H} x_i + \sum_{i \in H} Ber\left(\frac{1}{2}\right) = r\right] \cdot \left(\frac{\lambda}{n}\right)^{|H|} \left(1 - \frac{\lambda}{n}\right)^{n-|H|}
\end{aligned} \tag{1}$$

Let $\mathbf{G}$ denote the (random) set of people for whom $b_i = 1$ in P_λ. Notice that

$$\begin{aligned}
\mathbb{P}\left[\sum_{i=1}^{n} R_{n,\lambda}(x_i) = r\right] &= \sum_{G \subseteq [n]} \mathbb{P}\left[\sum_{i} R_{n,\lambda}(x_i) = r \cap \mathbf{G} = G\right] \\
&= \sum_{G \subseteq [n]} \mathbb{P}\left[\sum_{i \notin G} x_i + \sum_{i \in G} Ber\left(\frac{1}{2}\right) = r\right] \\
&\quad \cdot \left(\frac{\lambda}{n}\right)^{|G|} \left(1 - \frac{\lambda}{n}\right)^{n-|G|}
\end{aligned}$$

which is the same as (1). This concludes the proof. □

Now we establish that in order to demonstrate privacy of $P_{n,\lambda}$, it suffices to analyze C_λ.

Claim 13. *If C_λ is (ε, δ) differentially private, then $P_{n,\lambda}$ is (ε, δ) differentially private.*

Proof. Fix any number of users n. Consider the randomized algorithm $T : \{0, 1, 2, \dots, n\} \to \{0,1\}^n$ that takes a number r and outputs a uniformly random

string z that has r ones. If C_λ is differentially private, then the output of $T \circ C_\lambda$ is (ε, δ) differentially private by the post-processing lemma.

To complete the proof, we show that for any $X \in \mathcal{X}^n$ the output of $(T \circ C_\lambda)(X)$ has the same distribution as $S(R_\lambda(x_1), \dots R_\lambda(x_n))$. Fix some vector $Z \in \{0,1\}^n$ with sum r

$$\begin{aligned}
\mathop{\mathbb{P}}_{T, C_\lambda}[T(C_\lambda(X)) = Z] &= \mathbb{P}[T(r) = Z] \cdot \mathbb{P}[C_\lambda(X) = r] \\
&= \binom{n}{r}^{-1} \cdot \mathbb{P}[C_\lambda(X) = r] \\
&= \binom{n}{r}^{-1} \cdot \mathbb{P}[f(R_{n,\lambda}(X)) = r] && \text{(Claim 12)} \\
&= \binom{n}{r}^{-1} \cdot \sum_{Y \in \{0,1\}^n : |Y| = r} \mathbb{P}[R_{n,\lambda}(X) = Y] \\
&= \sum_{Y \in \{0,1\}^n : |Y| = r} \mathbb{P}[R_{n,\lambda}(X) = Y] \cdot \mathbb{P}[S(Y) = Z] \\
&= \mathop{\mathbb{P}}_{R_{n,\lambda}, S}[S(R_{n,\lambda}(X)) = Z]
\end{aligned}$$

This completes the proof of Claim 13. □

We will analyze the privacy of C_λ in three steps. First we show that for *any* sufficiently large H, the final step (encapsulated by Algorithm 3) will ensure differential privacy for some parameters. When then show that for *any* sufficiently large value s and H chosen *randomly* with $|H| = s$, the privacy parameters actually improve significantly in the regime where s is close to n; this sampling of H is performed by Algorithm 4. Finally, we show that when s is chosen *randomly* then s is sufficiently large with high probability.

Algorithm 3: $C_H(x_1 \dots x_n)$

Input: $(x_1 \dots x_n) \in \{0,1\}^n$, parameter $H \subseteq [n]$.
Output: $\mathbf{y}_H \in \{0, 1, 2, \dots, n\}$

Let $\mathbf{B} \leftarrow Bin\left(|H|, \frac{1}{2}\right)$
Return $\mathbf{y}_H \leftarrow \sum_{i \notin H} x_i + \mathbf{B}$

Claim 14. *For any $\delta > 0$ and any $H \subseteq [n]$ such that $|H| > 8 \log \frac{4}{\delta}$, C_H is $(\varepsilon, \frac{\delta}{2})$-differentially private for*

$$\varepsilon = \ln\left(1 + \sqrt{\frac{32 \log \frac{4}{\delta}}{|H|}}\right) < \sqrt{\frac{32 \log \frac{4}{\delta}}{|H|}}$$

Proof. Fix neighboring datasets $X \sim X' \in \{0,1\}^n$, any $H \subseteq [n]$ such that $|H| > 8 \log \frac{4}{\delta}$, and any $\delta > 0$. If the point at which X, X' differ lies within H, the

two distributions $C_H(X), C_H(X')$ are identical. Hence, without loss of generality we assume that $x_j = 0$ and $x'_j = 1$ for some $j \notin H$.

Define $u := \sqrt{\frac{1}{2}|H| \log \frac{4}{\delta}}$ and $I_u := \left(\frac{1}{2}|H| - u, \frac{1}{2}|H| + u\right)$ so that by Hoeffding's inequality, $\mathbb{P}[\mathbf{B} \in I_u] < \frac{1}{2}\delta$. For any $W \subseteq \{0, 1, 2, \ldots, n\}$ we have,

$$\begin{aligned}
\mathbb{P}[C_H(X) \in W] &= \mathbb{P}[C_H(X) \in W \cap \mathbf{B} \in I_u] + \mathbb{P}[C_H(X) \in W \cap \mathbf{B} \notin I_u] \\
&\leq \mathbb{P}[C_H(X) \in W \cap \mathbf{B} \in I_u] + \frac{1}{2}\delta \\
&= \sum_{r \in W \cap I_u} \mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x_i = r\right] + \frac{1}{2}\delta
\end{aligned}$$

Thus to complete the proof, it suffices to show that for any H and $r \in W \cap I_u$

$$\frac{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x_i = r\right]}{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x'_i = r\right]} \leq 1 + \sqrt{\frac{32 \log \frac{4}{\delta}}{|H|}} \tag{2}$$

Because $x_j = 0, x'_j = 1$ and $j \notin H$, we have $\sum_{i \notin H} x_i = \sum_{i \notin H} x'_i - 1$. Thus,

$$\begin{aligned}
\frac{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x_i = r\right]}{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x'_i = r\right]} &= \frac{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x'_i - 1 = r\right]}{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x'_i = r\right]} \\
&= \frac{\mathbb{P}\left[\mathbf{B} = \left(r - \sum_{i \notin H} x'_i\right) + 1\right]}{\mathbb{P}\left[\mathbf{B} = \left(r - \sum_{i \notin H} x'_i\right)\right]}
\end{aligned}$$

Now we define $k = r - \sum_{i \notin H} x'_i$ so that

$$\frac{\mathbb{P}\left[\mathbf{B} = \left(r - \sum_{i \notin H} x'_i\right) + 1\right]}{\mathbb{P}\left[\mathbf{B} = \left(r - \sum_{i \notin H} x'_i\right)\right]} = \frac{\mathbb{P}[\mathbf{B} = k + 1]}{\mathbb{P}[\mathbf{B} = k]}.$$

Then we can calculate

$$\begin{aligned}
\frac{\mathbb{P}[\mathbf{B} = k+1]}{\mathbb{P}[\mathbf{B} = k]} &= \frac{|H| - k}{k + 1} && (\mathbf{B} \text{ is binomial}) \\
&\leq \frac{|H| - (\frac{1}{2}|H| - u)}{\frac{1}{2}|H| - u + 1} && (r \in I_u \text{ so } k \geq \tfrac{1}{2}|H| - u) \\
&< \frac{\frac{1}{2}|H| + u}{\frac{1}{2}|H| - u} = \frac{u^2/(\log \frac{4}{\delta}) + u}{u^2/(\log \frac{4}{\delta}) - u} && (u = \sqrt{\tfrac{1}{2}|H| \log \tfrac{4}{\delta}}) \\
&= \frac{u + \log \frac{4}{\delta}}{u - \log \frac{4}{\delta}} = 1 + \frac{2 \log \frac{4}{\delta}}{u - \log \frac{4}{\delta}} = 1 + \frac{2 \log \frac{4}{\delta}}{\sqrt{\frac{1}{2}|H| \log \frac{4}{\delta}} - \log \frac{4}{\delta}} \\
&\leq 1 + \frac{4 \log \frac{4}{\delta}}{\sqrt{\frac{1}{2}|H| \log \frac{4}{\delta}}} = 1 + \sqrt{\frac{32 \log \frac{4}{\delta}}{|H|}} && (|H| > 8 \log \tfrac{4}{\delta})
\end{aligned}$$

which completes the proof. □

Next, we consider the case where H is a *random* subset of $[n]$ with a *fixed* size s. In this case we will use an *amplification via sampling argument* [20, 27] to argue that the randomness of H improves the privacy parameters by a factor of roughly $(1 - \frac{s}{n})$, which will be crucial when $s \approx n$.

Algorithm 4: $C_s(x_1, \ldots, x_n)$

Input: $(x_1, \ldots, x_n) \in \{0,1\}^n$, parameter $s \in \{0, 1, 2, \ldots, n\}$.
Output: $\mathbf{y}_s \in \{0, 1, 2, \ldots, n\}$

Define $\mathcal{H}_s = \{H \subseteq [n] : |H| = s\}$ and choose $\mathbf{H} \leftarrow \mathcal{H}_s$ uniformly at random
Return $\mathbf{y}_s \leftarrow C_{\mathbf{H}}(x)$

Claim 15. *For any $\delta > 0$ and any $s > 8 \log \frac{4}{\delta}$, C_s is $(\varepsilon, \frac{1}{2}\delta)$ differentially private for*

$$\varepsilon = \sqrt{\frac{32 \log \frac{4}{\delta}}{s}} \cdot \left(1 - \frac{s}{n}\right)$$

Proof. As in the previous section, fix $X \sim X' \in \{0,1\}^n$ where $x_j = 0, x'_j = 1$. $C_s(X)$ selects $\mathbf{H}$ uniformly from $\mathcal{H}_s$ and runs $C_H(X)$; let H denote the realization of $\mathbf{H}$. To enhance readability, we will use the shorthand $\varepsilon_0(s) := \sqrt{\frac{32 \log \frac{4}{\delta}}{s}}$. For any $W \subset \{0, 1, 2, \ldots, n\}$, we aim to show that

$$\frac{\mathop{\mathbb{P}}_{\mathbf{H}, C_{\mathbf{H}}}[C_{\mathbf{H}}(X) \in W] - \frac{1}{2}\delta}{\mathop{\mathbb{P}}_{\mathbf{H}, C_{\mathbf{H}}}[C_{\mathbf{H}}(X') \in W]} \leq \exp\left(\varepsilon_0(s) \cdot \left(1 - \frac{s}{n}\right)\right)$$

First, we have

$$\begin{aligned}
&\frac{\mathop{\mathbb{P}}_{\mathbf{H}, C_{\mathbf{H}}}[C_{\mathbf{H}}(X) \in W] - \frac{1}{2}\delta}{\mathop{\mathbb{P}}_{\mathbf{H}, C_{\mathbf{H}}}[C_{\mathbf{H}}(X') \in W]} \\
&= \frac{\mathop{\mathbb{P}}_{\mathbf{H}, C_{\mathbf{H}}}[C_{\mathbf{H}}(X) \in W \cap j \in \mathbf{H}] + \mathop{\mathbb{P}}_{\mathbf{H}, C_{\mathbf{H}}}[C_{\mathbf{H}}(X) \in W \cap j \notin \mathbf{H}] - \frac{1}{2}\delta}{\mathop{\mathbb{P}}_{\mathbf{H}, C_{\mathbf{H}}}[C_{\mathbf{H}}(X') \in W \cap j \in \mathbf{H}] + \mathop{\mathbb{P}}_{\mathbf{H}, C_{\mathbf{H}}}[C_{\mathbf{H}}(X') \in W \cap j \notin \mathbf{H}]} \\
&= \frac{(1-p)\gamma(X) + p\zeta(X) - \frac{1}{2}\delta}{(1-p)\gamma(X') + p\zeta(X')} \qquad (3)
\end{aligned}$$

where $p := \mathbb{P}[j \notin \mathbf{H}] = (1 - s/n)$,

$$\gamma(X) := \mathop{\mathbb{P}}_{C_{\mathbf{H}}}[C_{\mathbf{H}}(X) \in W \mid j \in \mathbf{H}] \quad \text{and} \quad \zeta(X) := \mathop{\mathbb{P}}_{C_{\mathbf{H}}}[C_{\mathbf{H}}(X) \in W \mid j \notin \mathbf{H}].$$

When user j outputs a uniformly random bit, their private value has no impact on the distribution. Hence, $\gamma(X) = \gamma(X')$, and

$$(3) = \frac{(1-p)\gamma(X) + p\zeta(X) - \frac{1}{2}\delta}{(1-p)\gamma(X) + p\zeta(X')} \qquad (4)$$

Since $s = |H|$ is sufficiently large, by Claim 14 we have $\zeta(X) \leq (1+\varepsilon_0(s)) \cdot \min\{\zeta(X'), \gamma(X)\} + \frac{1}{2}\delta$.

$$\begin{aligned}
(4) &\leq \frac{(1-p)\gamma(X) + p\cdot(1+\varepsilon_0(s))\cdot\min\{\zeta(X'),\gamma(X)\} + \delta) - \frac{1}{2}\delta}{(1-p)\gamma(X) + p\zeta(X')} \\
&\leq \frac{(1-p)\gamma(X) + p\cdot(1+\varepsilon_0(s))\cdot\min\{\zeta(X'),\gamma(X)\}}{(1-p)\gamma(X) + p\zeta(X')} \\
&= \frac{(1-p)\gamma(X) + p\cdot\min(\zeta(X'),\gamma(X)) + p\cdot\varepsilon_0(s)\cdot\min\{\zeta(X'),\gamma(X)\}}{(1-p)\gamma(X) + p\zeta(X')} \\
&\leq \frac{(1-p)\gamma(X) + p\zeta(X') + p\cdot\varepsilon_0(s)\cdot\min\{\zeta(X'),\gamma(X)\}}{(1-p)\gamma(X) + p\zeta(X')} \\
&= 1 + \frac{p\cdot\varepsilon_0(s)\cdot\min\{\zeta(X'),\gamma(X)\}}{(1-p)\gamma(X) + p\zeta(X')} \qquad (5)
\end{aligned}$$

Observe that $\min\{\zeta(X'),\gamma(X)\} \leq (1-p)\gamma(X) + p\zeta(X')$, so

$$\begin{aligned}
(5) \leq 1 + p\cdot\varepsilon_0(s) = 1 + \varepsilon_0(s)\cdot\left(1-\frac{s}{n}\right) &\leq \exp\left(\varepsilon_0(s)\cdot\left(1-\frac{s}{n}\right)\right) \\
&= \exp\left(\sqrt{\frac{32\log(4/\delta)}{s}}\cdot\left(1-\frac{s}{n}\right)\right)
\end{aligned}$$

which completes the proof. □

We now come to the actual algorithm C_λ, where s is not fixed but is random. The analysis of C_s yields a bound on the privacy parameter that increases with s, so we will complete the analysis of C_λ by using the fact that, with high probability, s is almost as large as λ.

Claim 16. *For any $\delta > 0$ and $n \geq \lambda \geq 14\log\frac{4}{\delta}$, C_λ is (ε,δ) differentially private where*

$$\varepsilon = \sqrt{\frac{32\log\frac{4}{\delta}}{\lambda - \sqrt{2\lambda\log\frac{2}{\delta}}}}\cdot\left(1 - \frac{\lambda - \sqrt{2\lambda\log\frac{2}{\delta}}}{n}\right)$$

The proof is in the full version of the paper

From Claim 13, C_λ and $P_{n,\lambda}$ share the same privacy guarantees. Hence, Claim 16 implies the following:

Corollary 17. *For any $\delta \in (0,1)$, $n \in \mathbb{N}$, and $\lambda \in \left[14\log\frac{4}{\delta}, n\right]$, $P_{n,\lambda}$ is (ε,δ) differentially private, where*

$$\varepsilon = \sqrt{\frac{32\log\frac{4}{\delta}}{\lambda - \sqrt{2\lambda\log\frac{2}{\delta}}}}\cdot\left(1 - \frac{\lambda - \sqrt{2\lambda\log\frac{2}{\delta}}}{n}\right)$$

4.3 Setting the Randomization Parameter

Corollary 17 gives a bound on the privacy of $P_{n,\lambda}$ in terms of the number of users n and the randomization parameter λ. While this may be enough on its own, in order to understand the tradeoff between ε and the accuracy of the protocol, we want to identify a suitable choice of λ to achieve a desired privacy guarantee (ε, δ). To complete the proof of Theorem 11, we prove such a bound.

For the remainder of this section, fix some $\delta \in (0,1)$. Corollary 17 states that for any n and $\lambda \in \left[14 \log \frac{4}{\delta}, n\right]$, $P_{n,\lambda}$ satisfies $(\varepsilon^*(\lambda), \delta)$-differential privacy, where

$$\varepsilon^*(\lambda) = \sqrt{\frac{32 \log \frac{4}{\delta}}{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}} \cdot \left(1 - \frac{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}{n}\right)$$

Let $\lambda^*(\varepsilon)$ be the inverse of ε^*, i.e. the minimum $\lambda \in [0, n]$ such that $\varepsilon^*(\lambda) \leq \varepsilon$. Note that $\varepsilon^*(\lambda)$ is decreasing as $\lambda \to n$ while $\lambda^*(\varepsilon)$ increases as $\varepsilon \to 0$. By definition, $P_{n,\lambda}$ satisfies (ε, δ) privacy if $\lambda \geq \lambda^*(\varepsilon)$; the following Lemma gives such an upper bound:

Lemma 18. *For all $\delta \in (0,1)$, $n \geq 14 \log \frac{4}{\delta}$, $\varepsilon \in \left(\frac{\sqrt{3456}}{n} \log \frac{4}{\delta}, 1\right)$, $P_{n,\lambda}$ is (ε, δ) differentially private if*

$$\lambda = \begin{cases} \frac{64}{\varepsilon^2} \log \frac{4}{\delta} & \text{if } \varepsilon \geq \sqrt{\frac{192}{n} \log \frac{4}{\delta}} \\ n - \frac{\varepsilon n^{3/2}}{\sqrt{432 \log(4/\delta)}} & \text{otherwise} \end{cases} \tag{6}$$

We'll prove the lemma in two claims, each of which corresponds to one of the two cases of our bound on $\lambda^*(\varepsilon)$. The first bound applies when ε is relatively large.

Claim 19. *For all $\delta \in (0,1)$, $n \geq 14 \log \frac{4}{\delta}$, $\varepsilon \in \left(\sqrt{\frac{192}{n} \log \frac{4}{\delta}}, 1\right)$, if $\lambda = \frac{64}{\varepsilon^2} \log \frac{4}{\delta}$ then $P_{n,\lambda}$ is (ε, δ) private.*

Proof. Let $\lambda = \frac{64}{\varepsilon^2} \log \frac{4}{\delta}$ as in the statement. Corollary 17 states that $P_{n,\lambda}$ satisfies $(\varepsilon^*(\lambda), \delta)$ privacy for

$$\begin{aligned} \varepsilon^*(\lambda) &= \sqrt{\frac{32 \log \frac{4}{\delta}}{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}} \cdot \left(1 - \frac{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}{n}\right) \\ &\leq \sqrt{\frac{32 \log \frac{4}{\delta}}{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}} && (\lambda \leq n) \\ &\leq \sqrt{\frac{64 \log \frac{4}{\delta}}{\lambda}} && (\lambda \geq 8 \log \tfrac{2}{\delta}) \\ &= \varepsilon \end{aligned}$$

This completes the proof of the claim. □

The value of λ in the previous claim can be as large as n when ε approaches $1/\sqrt{n}$. We now give a meaningful bound for smaller values of ε.

Claim 20. *For all $\delta \in (0,1)$, $n \geq 14 \log \frac{4}{\delta}$, $\varepsilon \in \left(\frac{\sqrt{3456}}{n} \log \frac{4}{\delta}, \sqrt{\frac{192}{n} \log \frac{4}{\delta}}\right)$, if*

$$\lambda = n - \frac{\varepsilon n^{3/2}}{\sqrt{432 \log(4/\delta)}}$$

then $P_{n,\lambda}$ is (ε, δ) private.

Proof. Let $\lambda = n - \varepsilon n^{3/2}/\sqrt{432 \log(4/\delta)}$ as in the statement. Note that for this ε regime, we have $n/3 < \lambda < n$. Corollary 17 states that $P_{n,\lambda}$ satisfies $(\varepsilon^*(\lambda), \delta)$ privacy for

$$\begin{aligned}
\varepsilon^*(\lambda) &= \sqrt{\frac{32 \log \frac{4}{\delta}}{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}} \cdot \left(1 - \frac{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}{n}\right) \\
&\leq \sqrt{\frac{64 \log \frac{4}{\delta}}{\lambda}} \cdot \left(1 - \frac{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}{n}\right) && (\lambda \geq 8 \log \tfrac{2}{\delta}) \\
&= \sqrt{\frac{64 \log \frac{4}{\delta}}{\lambda}} \cdot \left(\frac{\varepsilon \sqrt{n}}{\sqrt{432 \log(4/\delta)}} + \frac{\sqrt{2\lambda \log \frac{2}{\delta}}}{n}\right) \\
&\leq \sqrt{\frac{64 \log \frac{4}{\delta}}{\lambda}} \cdot \left(\frac{\varepsilon \sqrt{n}}{\sqrt{432 \log(4/\delta)}} + \sqrt{\frac{2 \log \frac{2}{\delta}}{n}}\right) && (\lambda \leq n) \\
&\leq \sqrt{\frac{192 \log \frac{4}{\delta}}{n}} \cdot \left(\frac{\varepsilon \sqrt{n}}{\sqrt{432 \log(4/\delta)}} + \sqrt{\frac{2 \log \frac{2}{\delta}}{n}}\right) && (\lambda \geq n/3) \\
&= \frac{2}{3}\varepsilon + \frac{\sqrt{384 \log \frac{4}{\delta} \log \frac{2}{\delta}}}{n} < \frac{2}{3}\varepsilon + \frac{\sqrt{384}}{n} \log \frac{4}{\delta} \\
&< \frac{2}{3}\varepsilon + \frac{1}{3}\varepsilon = \varepsilon && (\varepsilon > \tfrac{\sqrt{3456}}{n} \log \tfrac{4}{\delta})
\end{aligned}$$

which completes the proof. □

4.4 Accuracy Analysis

In this section, we will bound the error of $P_\lambda(X)$ with respect to $\sum_i x_i$. Recall that, to clean up notational clutter, we will often write $f(X) = \sum_i x_i$. As with the previous section, our statements will at first be in terms of λ but the section will end with a statement in terms of ε, δ.

Theorem 21. *For every $n \in \mathbb{N}$, $\beta > 0$, $n > \lambda \geq 2 \log \frac{2}{\beta}$, and $x \in \{0,1\}^n$,*

$$\mathbb{P}\left[\left|P_{n,\lambda}(x) - \sum_i x_i\right| > \sqrt{2\lambda \log(2/\beta)} \cdot \left(\frac{n}{n-\lambda}\right)\right] \leq \beta$$

Observe that, using the choice of λ specified in Theorem 11, we conclude that for every $\frac{1}{n} \lesssim \varepsilon \lesssim 1$ and every δ the protocol P_λ satisfies

$$\mathbb{P}\left[\left|P_{n,\lambda}(x) - \sum_i x_i\right| > O\left(\frac{\sqrt{\log(1/\delta)\log(1/\beta)}}{\varepsilon}\right)\right] \leq \beta$$

To see how this follows from Theorem 21, consider two parameter regimes:

1. When $\varepsilon \gg 1/\sqrt{n}$ then $\lambda \approx \frac{\sqrt{\log(1/\delta)}}{\varepsilon^2} \ll n$, so the bound in Theorem 21 is $O(\sqrt{\lambda \log(1/\beta)})$, which yields the desired bound.
2. When $\varepsilon \ll 1/\sqrt{n}$ then $n - \lambda \approx \varepsilon n^{3/2}/\sqrt{\log(1/\delta)} \ll n$, so the bound in Theorem 21 is $O\left(\frac{n^{3/2}\sqrt{\log(1/\beta)}}{n-\lambda}\right)$, which yields the desired bound.

Theorem 2 in the introduction follows from this intuition; a formal proof can be found in the full version.

5 A Protocol for Sums of Real Numbers

In this section, we show how to extend our protocol to compute sums of bounded real numbers. In this case the data domain is $\mathcal{X} = [0,1]$, but the function we wish to compute is still $f(x) = \sum_i x_i$. The main idea of the protocol is to randomly round each number x_i to a Boolean value $b_i \in \{0,1\}$ with expected value x_i. However, since the randomized rounding introduces additional error, we may need to round multiple times and estimate several sums. As a consequence, this protocol is not one-message.

5.1 The Protocol

Our algorithm is described in two parts, an encoder E_r that performs the randomized rounding (Algorithm 5) and a shuffled protocol $P^{\mathbb{R}}_{\lambda,r}$ (Algorithm 6) that is the composition of many copies of our protocol for the binary case, $P^{0/1}_\lambda$. The encoder takes a number $x \in [0,1]$ and a parameter $r \in \mathbb{N}$ and outputs a vector $(b_1, \ldots, b_r) \in \{0,1\}^r$ such that $\mathbb{E}\left[\frac{1}{r}\sum_j b_j\right] = x_j$ and $\mathrm{Var}\left[\frac{1}{r}\sum_j b_j\right] = O(1/r^2)$. To clarify, we give two examples of the encoding procedure:

- If $r = 1$ then the encoder simply sets $b = Ber(x)$. The mean and variance of b are x and $x(1-x) \leq \frac{1}{4}$, respectively.
- If $x = .4$ and $r = 4$ then the encoder sets $b = (1, Ber(.6), 0, 0)$. The mean and variance of $\frac{1}{4}(b_1 + b_2 + b_3 + b_4)$ are .4 and .015, respectively.

After doing the rounding, we then run the bit-sum protocol $P_\lambda^{0/1}$ on the bits $b_{1,j}, \ldots, b_{n,j}$ for each $j \in [r]$ and average the results to obtain an estimate of the quantity

$$\sum_i \frac{1}{r} \sum_j b_{i,j} \approx \sum_i x_i$$

To analyze privacy we use the fact that the protocol is a composition of bit-sum protocols, which are each private, and thus we can analyze privacy via the composition properties of differential privacy.

Much like in the bit-sum protocol, we use $P^{\mathbb{R}}_{n,\lambda,r}, R^{\mathbb{R}}_{n,\lambda,r}, A^{\mathbb{R}}_{n,\lambda,r}$ to denote the real-sum protocol and its components when n users participate.

Algorithm 5: An encoder $E_r(x)$

Input: $x \in [0,1]$, a parameter $r \in \mathbb{N}$.
Output: $(\mathbf{b}_1, \ldots, \mathbf{b}_r) \in \{0,1\}^r$

Let $\mu \leftarrow \lceil x \cdot r \rceil$ and $p \leftarrow x \cdot r - \mu + 1$
For $j = 1, \ldots, r$

$$\mathbf{b}_j = \begin{cases} 1 & j < \mu \\ Ber(p) & j = \mu \\ 0 & j > \mu \end{cases}$$

Return $(\mathbf{b}_1, \ldots, \mathbf{b}_r)$

Algorithm 6: The protocol $P^{\mathbb{R}}_{\lambda,r} = (R^{\mathbb{R}}_{\lambda,r}, S, A^{\mathbb{R}}_{\lambda,r})$

`// Local randomizer`
$R^{\mathbb{R}}_{n,\lambda,r}(x)$**:**
Input: $x \in [0,1]$, parameters $n, r \in \mathbb{N}, \lambda \in (0,n)$.
Output: $(\mathbf{y}_1, \ldots \mathbf{y}_r) \in \{0,1\}^r$
$(\mathbf{b}_1, \ldots \mathbf{b}_r) \leftarrow E_r(x)$
Return $(\mathbf{y}_1, \ldots \mathbf{y}_r) \leftarrow \left(R^{0/1}_{n,\lambda}(\mathbf{b}_1), \ldots, R^{0/1}_{n,\lambda}(\mathbf{b}_r)\right)$

`// Analyzer`
$A^{\mathbb{R}}_{n,\lambda,r}(y_{1,1}, \ldots, y_{n,r})$**:**
Input: $(y_{1,1}, \ldots, y_{n,r}) \in \{0,1\}^{n \cdot r}$, parameters $n, r \in \mathbb{N}, \lambda \in (0,n)$.
Output: $z \in [0,n]$
Return $z \leftarrow \frac{1}{r} \cdot \frac{n}{n-\lambda} \left(\left(\sum_j \sum_i y_{i,j} \right) - \frac{\lambda \cdot r}{2} \right)$

Theorem 22. *For every $\delta = \delta(n)$ such that $e^{-\Omega(n^{1/4})} < \delta(n) < \frac{1}{n}$ and $\frac{\text{poly}(\log n)}{n} < \varepsilon < 1$ and every sufficiently large n, there exists parameters $\lambda \in [0, n], r \in \mathbb{N}$ such that $P^{\mathbb{R}}_{n,\lambda,r}$ is both (ε, δ) differentially private and for every $\beta > 0$, and every $X = (x_1, \ldots, x_n) \in [0, 1]^n$,*

$$\mathbb{P}\left[\left|P^{\mathbb{R}}_{n,\lambda,r}(X) - \sum_{i=1}^{n} x_i\right| > O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\sqrt{\log\frac{1}{\beta}}\right)\right] \leq \beta$$

5.2 Privacy Analysis

Privacy will follow immediately from the composition properties of shuffled protocols (Lemma 10) and the privacy of the bit-sum protocol $P_{n,\lambda}$. One technical nuisance is that the composition properties are naturally stated in terms of ε, whereas the protocol is described in terms of the parameter λ, and the relationship between ε, λ, and n is somewhat complex. Thus, we will state our guarantees in terms of the level of privacy that each individual bit-sum protocol achieves with parameter λ. To this end, define the function $\lambda^*(n, \varepsilon, \delta)$ to be the minimum value of λ such that the bit-sum protocol with n users satisfies (ε, δ)-differential privacy. We will state the privacy guarantee in terms of this function.

Theorem 23. *For every $\varepsilon, \delta \in (0, 1), n, r \in \mathbb{N}$, define*

$$\varepsilon_0 = \frac{\varepsilon}{\sqrt{8r\log(2/\delta)}} \qquad \delta_0 = \frac{\delta}{2r} \qquad \lambda^* = \lambda^*(n, \varepsilon_0, \delta_0)$$

For every $\lambda \geq \lambda^*$, $P^{\mathbb{R}}_{n,\lambda,r}$ is (ε, δ)-differentially private.

5.3 Accuracy Analysis

In this section, we bound the error of $P^{\mathbb{R}}_{\lambda,r}(X)$ with respect to $\sum_i x_i$. Recall that $f(X) = \sum_i x_i$.

Observe that there are two sources of randomness: the encoding of the input $X = (x_1, \ldots x_n)$ as bits and the execution of $R^{0/1}_{n,\lambda}$ on that encoding. We first show that the bit encoding lends itself to an unbiased and concentrated estimator of $f(X)$. Then we show that the output of $P_{n,\lambda,r}$ is concentrated around any value that estimator takes.

Theorem 24. *For every $\beta > 0$, $n \geq \lambda \geq \frac{16}{9}\log\frac{2}{\beta}$, $r \in \mathbb{N}$, and $X \in [0, 1]^n$,*

$$\mathbb{P}\left[\left|P^{\mathbb{R}}_{n,\lambda,r}(X) - f(X)\right| \geq \frac{\sqrt{2}}{r}\sqrt{n\log\frac{2}{\beta}} + \frac{n}{n-\lambda}\cdot\sqrt{2\frac{\lambda}{r}\log\frac{2}{\beta}}\right] < 2\beta$$

The analysis can be found in the full version of the paper, which also argues that setting $r \leftarrow \varepsilon \cdot \sqrt{n}$ suffices to achieve the bound in Theorem 22.

6 Lower Bounds for the Shuffled Model

In this section, we prove separations between central model algorithms and shuffled model protocols where each user's local randomizer is identical and sends one indivisible message to the shuffler (the one-message model).

Theorem 25 (Shuffled-to-Local Transformation). *Let P_S be a protocol in the one-message shuffled model that is*

- *$(\varepsilon_S, \delta_S)$-differentially private in the shuffled model for some $\varepsilon_S \leq 1$ and $\delta_S = \delta_S(n) < n^{-8}$, and*
- *(α, β)-accurate with respect to f for some $\beta = \Omega(1)$.*

Then there exists a protocol P_L in the local model that is

- *$(\varepsilon_L, 0)$-differentially private in the local model for $\varepsilon_L = 8(\varepsilon_S + \ln n)$, and*
- *$(\alpha, 4\beta)$-accurate with respect to f (when n is larger than some absolute constant)*

This means that an impossibility result for approximating f in the local model implies a related impossibility result for approximating f in the shuffled model. In Section 6.2 we combine this result with existing lower bounds for local differential privacy to obtain several strong separations between the central model and the one-message shuffled model.

The key to Theorem 25 is to show that if $P_S = (R_S, S, A_S)$ is a protocol in the one-message shuffled model satisfying $(\varepsilon_S, \delta_S)$-differential privacy, then the algorithm R_S itself satisfies $(\varepsilon_L, \delta_S)$-differential privacy without use of the shuffler S. Therefore, the local protocol $P_L = (R_S, A_S \circ S)$ is $(\varepsilon_L, \delta_S)$-private in the local model and has the exact same output distribution, and thus the exact same accuracy, as P_S. To complete the proof, we use (a slight generalization of) a transformation of Bun, Nelson, and Stemmer [7] to turn R into a related algorithm R' satisfying $(8(\varepsilon_S + \ln n), 0)$-differential privacy with only a slight loss of accuracy. We prove the latter result in the full version of the paper.

6.1 One-message Randomizers Satisfy Local Differential Privacy

The following lemma is the key step in the proof of Theorem 25, and states that for any symmetric shuffled protocol, the local randomizer R must satisfy local differential privacy with weak, but still non-trivial, privacy parameters.

Theorem 26. *Let $P = (R, S, A)$ be a protocol in the one-message shuffled model. If $n \in \mathbb{N}$ is such that P_n satisfies $(\varepsilon_S, \delta_S)$-differential privacy, then the algorithm R satisfies $(\varepsilon_L, \delta_L)$-differential privacy for $\varepsilon_L = \varepsilon_S + \ln n$. Therefore, the symmetric local protocol $P_L = (R, A \circ S)$ satisfies $(\varepsilon_L, \delta_L)$-differential privacy.*

Proof. By assumption, P_n is $(\varepsilon_S, \delta_S)$-private. Let ε be the supremum such that $R : \mathcal{X} \to \mathcal{Y}$ is *not* (ε, δ_S)-private. We will attempt to find a bound on ε. If R is not (ε, δ_S)-differentially private, there exist $Y \subset \mathcal{Y}$ and $x, x' \in \mathcal{X}$ such that

$$\mathbb{P}\left[R(x') \in Y\right] > \exp(\varepsilon) \cdot \mathbb{P}\left[R(x) \in Y\right] + \delta_S$$

For brevity, define $p := \mathbb{P}(R(x) \in Y)$ and $p' := \mathbb{P}(R(x') \in Y)$ so that we have

$$p' > \exp(\varepsilon)p + \delta_S \tag{7}$$

We will show that if ε is too large, then (7) will imply that P_n is *not* $(\varepsilon_S, \delta_S)$-differentially private, which contradicts our assumption. To this end, define the set $\mathcal{W} := \{W \in \mathcal{Y}^n \mid \exists i \; w_i \in Y\}$. Define two datasets $X \sim X'$ as

$$X := (\underbrace{x, \ldots, x}_{n \text{ times}}) \quad \text{and} \quad X' := (x', \underbrace{x, \ldots, x}_{n-1 \text{ times}})$$

Because P_n is $(\varepsilon_S, \delta_S)$-differentially private

$$\mathbb{P}\left[P_n(X') \in \mathcal{W}\right] \leq \exp(\varepsilon_S) \cdot \mathbb{P}\left[P_n(X) \in W\right] + \delta_S \tag{8}$$

Now we have

$$\begin{aligned}
&\mathbb{P}\left[P_n(X) \in \mathcal{W}\right] \\
&= \mathbb{P}\left[S(\underbrace{R(x), \ldots, R(x)}_{n \text{ times}}) \in \mathcal{W}\right] \\
&= \mathbb{P}\left[(\underbrace{R(x), \ldots, R(x)}_{n \text{ times}}) \in \mathcal{W}\right] && (\mathcal{W} \text{ is symmetric}) \\
&= \mathbb{P}\left[\exists i \;\; R(x) \in Y\right] \leq n \cdot \mathbb{P}\left[R(x) \in Y\right] && (\text{Union bound}) \\
&= np
\end{aligned}$$

where the second equality is because the set W is closed under permutation, so we can remove the random permutation S without changing the probability. Similarly, we have

$$\begin{aligned}
\mathbb{P}\left[P_n(X') \in \mathcal{W}\right] = \mathbb{P}\left[(R(x'), \underbrace{R(x) \ldots, R(x)}_{n-1 \text{ times}}) \in \mathcal{W}\right] \\
\geq \mathbb{P}\left[R(x') \in Y\right] = p' \\
> \exp(\varepsilon)p + \delta_S && (\text{By } (7))
\end{aligned}$$

Now, plugging the previous two inequalities into (8), we have

$$\begin{aligned}
\exp(\varepsilon)p + \delta_S &< \mathbb{P}\left[P_n(X') \in \mathcal{W}\right] \\
&\leq \exp(\varepsilon_S) \cdot \mathbb{P}\left[P_n(X) \in \mathcal{W}\right] \\
&\leq \exp(\varepsilon_S)np + \delta_S
\end{aligned}$$

By rearranging and canceling terms in the above we obtain the conclusion

$$\varepsilon \leq \varepsilon_S + \ln n$$

Therefore R must satisfy $(\varepsilon_S + \ln n, \delta_S)$-differential privacy. □

Claim 27. *If the shuffled protocol $P_S = (R, S, A)$ is (α, β)-accurate for some function f, then the local protocol $P_L = (R, A \circ S)$ is (α, β)-accurate for f, where*

$$(A \circ S)(y_1, \ldots, y_N) = A(S(y_1, \ldots, y_N))$$

We do not present a proof of Claim 27, as it is immediate that the distribution of $P_S(x)$ and $P_L(x)$ are identical, since $A \circ S$ incorporates the shuffler.

We conclude this section with a slight extension of a result of Bun, Nelson, and Stemmer [7] showing how to transform any local algorithm satisfying (ε, δ)-differential privacy into one satisfying $(O(\varepsilon), 0)$-differential privacy with only a small decrease in accuracy. Our extension covers the case where $\varepsilon > 2/3$, whereas their result as stated requires $\varepsilon \leq 1/4$.

Theorem 28 (Extension of [7]). *Suppose local protocol $P_L = (R, A)$ is (ε, δ) differentially private and (α, β) accurate with respect to f. If $\varepsilon > 2/3$ and*

$$\delta < \frac{\beta}{8n \ln(n/\beta)} \cdot \frac{1}{\exp(6\varepsilon)}$$

then there exists another local protocol $P'_L = (R', A)$ that is $(8\varepsilon, 0)$ differentially private and $(\alpha, 4\beta)$ accurate with respect to f.

The proof can be found in the full version of the paper. Theorem 25 now follows by combining Theorem 26 and Claim 27 with Theorem 28.

6.2 Applications of Theorem 25

In this section, we define two problems and present known lower bounds in the central and local models. By applying Theorem 25, we derive lower bounds in the one-message shuffled model. These bounds imply large separations between the central and one-message shuffled models.

The Selection Problem We define the *selection problem* as follows. The data universe is $\mathcal{X} = \{0,1\}^d$ where d is the *dimension* of the problem and the main parameter of interest. Given a dataset $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$, the goal is to identify a coordinate j such that the sum of the users' j-th bits is approximately as large as possible. That is, a coordinate $j \in [d]$ such that

$$\sum_{i=1}^{n} x_{i,j} \geq \max_{j' \in [d]} \sum_{i=1}^{n} x_{i,j'} - \frac{n}{10} \tag{9}$$

We say that an algorithm *solves the selection problem with probability* $1 - \beta$ if for every dataset x, with probability at least $1 - \beta$, it outputs j satisfying (9).

We would like to understand the minimum n (as a function of d) such that there is a differentially private algorithm that can solve the selection problem with constant probability of failure. We remark that this is a very weak notion

Table 1: Comparisons Between Models. When a parameter is unspecified, the reader may substitute $\varepsilon = 1, \delta = 0, \alpha = \beta = .01$. **All results are presented as the minimum dataset size n for which we can hope to achieve the desired privacy and accuracy as a function of the relevant parameter for the problem.**

Function (Parameters)	Differential Privacy Model			
	Central	**Shuffled (this paper)**		Local
		One-Message	**General**	
Mean, $\mathcal{X} = \{0,1\}$ (Accuracy α)	$\Theta(\frac{1}{\alpha\varepsilon})$	$O(\frac{\sqrt{\log(1/\delta)}}{\alpha\varepsilon})$		$\Theta(\frac{1}{\alpha^2\varepsilon^2})$
Mean, $\mathcal{X} = [0,1]$ (Accuracy α)		$O(\frac{1}{\alpha^2} + \frac{\sqrt{\log(1/\delta)}}{\alpha\varepsilon})$	$O(\frac{\log(1/\delta)}{\alpha\varepsilon})$	
Selection (Dimension d)	$\Theta(\log d)$	$\Omega(d^{\frac{1}{17}})$	$\tilde{O}(\sqrt{d}\log\frac{d}{\delta})$	$\Theta(d\log d)$
Histograms (Domain Size D)	$\Theta\big(\min\{\log\frac{1}{\delta}, \log D\}\big)$	$\Omega(\log^{\frac{1}{17}} D)$	$O(\sqrt{\log D})$	$\Theta(\log D)$

of accuracy, but since we are proving a negative result, using a weak notion of accuracy only strengthens our results.

The following lower bound for locally differentially private protocols for selection is from [31], and is implicit in the work of [12].[2]

Theorem 29. *If $P_L = (R_L, A_L)$ is a local protocol that satisfies $(\varepsilon, 0)$-differential privacy and P_L solves the selection problem with probability $\frac{9}{10}$ for datasets $x \in (\{0,1\}^d)^n$, then $n = \Omega\left(\frac{d\log d}{(e^\varepsilon - 1)^2}\right)$.*

By applying Theorem 25 we immediately obtain the following corollary.

Corollary 30. *If $P_S = (R_S, S, A_S)$ is a $(1, \delta)$-differentially private protocol in the one-message shuffled model, for $\delta = \delta(n) < n^{-8}$, and P_S solves the selection problem with probability $\frac{99}{100}$, then $n = \Omega((d\log d)^{1/17})$.*

Using a multi-message shuffled protocol[3], we can solve selection with $\tilde{O}(\frac{1}{\varepsilon}\sqrt{d})$ samples. By contrast, in the local model $n = \Theta(\frac{1}{\varepsilon^2} d\log d)$ samples are necessary and sufficient. In the central model, this problem is solved by the *exponential mechanism* [25] with a dataset of size just $n = O(\frac{1}{\varepsilon}\log d)$, and this is optimal [2, 28]. These results are summarized in Table 1.

[2] These works assume that the dataset x consists of independent samples from some distribution $\mathcal{D}$, and define accuracy for selection with respect to mean of that distribution. By standard arguments, a lower bound for the distributional version implies a lower bound for the version we have defined.

[3] The idea is to simulate multiple rounds of our protocol for binary sums, one round per dimension.

Histograms We define the *histogram problem* as follows. The data universe is $\mathcal{X} = [D]$ where D is the *domain size* of the problem and the main parameter of interest. Given a dataset $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$, the goal is to build a vector of size D such that for all $j \in [D]$ the j-th element is as close to the frequency of j in x. That is, a vector $v \in [0, n]^D$ such that

$$\max_{j \in [D]} \left| v_j - \sum_{i=1}^{n} \mathbb{1}(x_i = j) \right| \leq \frac{n}{10} \tag{10}$$

where $\mathbb{1}(\texttt{conditional})$ is defined to be 1 if `conditional` evaluates to `true` and 0 otherwise.

Similar to the selection problem, an algorithm *solves the histogram problem with probability* $1 - \beta$ if for every dataset x, with probability at least $1 - \beta$ it outputs v satisfying (10). We would like to find the minimum n such that a differentially private algorithm can solve the histogram problem; the following lower bound for locally differentially private protocols for histograms is from [3].

Theorem 31. *If $P_L = (R_L, A_L)$ is a local protocol that satisfies $(\varepsilon, 0)$ differential privacy and P_L solves the histogram problem with probability $\frac{9}{10}$ for any $x \in [D]^n$ then $n = \Omega\left(\frac{\log D}{(e^\varepsilon - 1)^2}\right)$*

By applying Theorem 25, we immediately obtain the following corollary.

Corollary 32. *If $P_S = (R_S, S, A_S)$ is a $(1, \delta)$-differentially private protocol in the one-message shuffled model, for $\delta = \delta(n) < n^{-8}$, and P_S solves the histogram problem with probability $\frac{99}{100}$, then $n = \Omega\left(\log^{1/17} D\right)$*

In the shuffled model, we can solve this problem using our protocol for bit-sums by having each user encode their data as a "histogram" of just their value $x_i \in [D]$ and then running the bit-sum protocol D times, once for each value $j \in [D]$, which incurs error $O(\frac{1}{\varepsilon}\sqrt{\log \frac{1}{\delta} \log D})$.[4] But in the central model, this problem can be solved to error $O(\min\{\log \frac{1}{\delta}, \log D\})$, which is optimal (see, e.g. [32]). Thus, the central and one-message shuffled models are qualitatively different with respect to computing histograms: D may be infinite in the former whereas D must be bounded in the latter.

Acknowledgements

AC was supported by NSF award CCF-1718088. AS was supported by NSF awards IIS-1447700 and AF-1763786 and a Sloan Foundation Research Award. JU was supported by NSF awards CCF-1718088, CCF-1750640, CNS-1816028 and a Google Faculty Research Award.

[4] Note that changing one user's data can only change two entries of their local histogram, so we only have to scale ε, δ by a factor of 2 rather than a factor that grows with D.

Bibliography

[1] Abowd, J.M.: The U.S. Census Bureau adopts differential privacy. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2867–2867. KDD '18, ACM, New York, NY, USA (2018)

[2] Bafna, M., Ullman, J.: The price of selection in differential privacy. In: Conference on Learning Theory. pp. 151–168 (2017)

[3] Bassily, R., Smith, A.: Local, private, efficient protocols for succinct histograms. In: Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing. pp. 127–135. ACM (2015)

[4] Beimel, A., Nissim, K., Omri, E.: Distributed private data analysis: Simultaneously solving how and what. In: Annual International Cryptology Conference. pp. 451–468. Springer (2008)

[5] Bittau, A., Erlingsson, U., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., Seefeld, B.: PROCHLO: Strong privacy for analytics in the crowd. In: Proceedings of the Symposium on Operating Systems Principles (SOSP) (2017)

[6] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy preserving machine learning. IACR Cryptology ePrint Archive (2017)

[7] Bun, M., Nelson, J., Stemmer, U.: Heavy hitters and the structure of local privacy. In: ACM SIGMOD/PODS Conference International Conference on Management of Data (PODS 2018) (2018)

[8] Chan, T.H., Shi, E., Song, D.: Optimal lower bound for differentially private multi-party aggregation. In: Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings. pp. 277–288 (2012)

[9] Chan, T.H.H., Shi, E., Song, D.: Privacy-preserving stream aggregation with fault tolerance. In: Financial Cryptography. pp. 200–214 (2012)

[10] Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–90 (Feb 1981)

[11] Corrigan-Gibbs, H., Boneh, D.: Prio: Private, robust, and scalable computation of aggregate statistics. In: Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation. pp. 259–282. NSDI'17, USENIX Association, Berkeley, CA, USA (2017)

[12] Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Local privacy and statistical minimax rates. In: Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on. pp. 429–438. IEEE (2013)

[13] Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: EUROCRYPT (2006)

[14] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference (TCC) (2006)
[15] Dwork, C., Rothblum, G.N., Vadhan, S.P.: Boosting and differential privacy. In: FOCS. pp. 51–60. IEEE (2010)
[16] Erlingsson, U., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., Thakurta, A.: Amplification by shuffling: From local to central differential privacy by anonymity. In: Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '19 (2019)
[17] Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In: ACM Conference on Computer and Communications Security (CCS) (2014)
[18] Evfimievski, A., Gehrke, J., Srikant, R.: Limiting privacy breaches in privacy preserving data mining. In: PODS. pp. 211–222. ACM (2003)
[19] van den Hooff, J., Lazar, D., Zaharia, M., Zeldovich, N.: Vuvuzela: Scalable private messaging resistant to traffic analysis. In: Proceedings of the 25th Symposium on Operating Systems Principles. pp. 137–152. SOSP '15, ACM, New York, NY, USA (2015)
[20] Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.: What can we learn privately? In: Foundations of Computer Science (FOCS). IEEE (2008)
[21] Kasiviswanathan, S.P., Smith, A.: On the 'semantics' of differential privacy: A bayesian formulation. CoRR **arXiv:0803.39461 [cs.CR]** (2008)
[22] Kearns, M.J.: Efficient noise-tolerant learning from statistical queries. In: STOC. pp. 392–401. ACM (May 16-18 1993)
[23] Kwon, A., Lazar, D., Devadas, S., Ford, B.: Riffle: An efficient communication system with strong anonymity. PoPETs **2016**(2), 115–134 (2016)
[24] McMillan, R.: Apple tries to peek at user habits without violating privacy. The Wall Street Journal (2016)
[25] McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: IEEE Foundations of Computer Science (FOCS) (2007)
[26] Shi, E., Chan, T.H., Rieffel, E.G., Chow, R., Song, D.: Privacy-preserving aggregation of time-series data. In: Proceedings of the Network and Distributed System Security Symposium, (NDSS) 2011 (2011)
[27] Smith, A.: Differential privacy and the secrecy of the sample (2009)
[28] Steinke, T., Ullman, J.: Tight lower bounds for differentially private selection. In: Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on. pp. 552–563. IEEE (2017)
[29] Thakurta, A.G., Vyrros, A.H., Vaishampayan, U.S., Kapoor, G., Freudiger, J., Sridhar, V.R., Davidson, D.: Learning new words (May 9 2017), uS Patent 9,645,998
[30] Tyagi, N., Gilad, Y., Leung, D., Zaharia, M., Zeldovich, N.: Stadium: A distributed metadata-private messaging system. In: Proceedings of the 26th Symposium on Operating Systems Principles. pp. 423–440. SOSP '17, ACM, New York, NY, USA (2017)

[31] Ullman, J.: Tight lower bounds for locally differentially private selection. CoRR **abs/1802.02638** (2018)
[32] Vadhan, S.: The complexity of differential privacy. http://privacytools.seas.harvard.edu/publications /complexity-differential-privacy (2016)
[33] Warner, S.L.: Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association **60**(309), 63–69 (1965)