Journal of Privacy and Confidentiality
Vol. 11 (1) 2021
TPDP 2019
Submitted July 2020
Published Jan 2021

MANIPULATION ATTACKS IN LOCAL DIFFERENTIAL PRIVACY

ALBERT CHEU, ADAM SMITH, AND JONATHAN ULLMAN

Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA.
e-mail address: cheu.a@northeastern.edu

Department of Computer Science, Boston University, Boston, MA, USA.
e-mail address: ads22@bu.edu

Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA.
e-mail address: jullman@ccs.neu.edu

Abstract. Local differential privacy is a widely studied restriction on distributed algorithms that collect aggregates about sensitive user data, and is now deployed in several large systems. We initiate a systematic study of a fundamental limitation of locally differentially private protocols: *they are highly vulnerable to adversarial manipulation.* While any algorithm can be manipulated by adversaries who lie about their inputs, we show that any noninteractive locally differentially private protocol can be manipulated to a much greater extent—when the privacy level is high, or the domain size is large, a small fraction of users in the protocol can completely obscure the distribution of the honest users' input. We also construct protocols that are optimally robust to manipulation for a variety of common tasks in local differential privacy. Finally, we give simple experiments validating our theoretical results, and demonstrating that protocols that are optimal without manipulation can have dramatically different levels of robustness to manipulation. Our results suggest caution when deploying local differential privacy and reinforce the importance of efficient cryptographic techniques for the distributed emulation of centrally differentially private mechanisms.

1. Introduction

Many companies rely on aggregates and models computed on sensitive user data. The past few years have seen a wave of deployments of systems for collecting sensitive user data via *local differential privacy* (Evfimievski et al., 2003), notably Google's RAPPOR (Erlingsson et al., 2014) and Apple's deployment in *iOS* (Apple Differential Privacy Team, 2017). These protocols satisfy differential privacy (Dwork et al., 2006b), a widely studied restriction that limits the information leaked due to any one user's presence in the data. Furthermore, the privacy guarantee is enforced *locally*, by a user's device, without reliance on the correctness of other parts of the system. See Figure 1 for a diagram.

Key words and phrases: Differential Privacy.

* A version of this work appeared in the IEEE Symposium on Security and Privacy, 2021. This work can also be found at https://arxiv.org/abs/1909.09630.

www.journalprivacyconfidentiality.org
DOI:10.29012/jpc.754

Local differential privacy is attractive for deployments for several reasons. The trust assumptions are relatively weak and easily explainable to novice users. In contrast to centralized differential privacy, the data collector never collects raw data, reducing the legal, ethical, and technical burden of safeguarding the data. Moreover, local protocols are typically simple and highly efficient in terms of communication and computation.

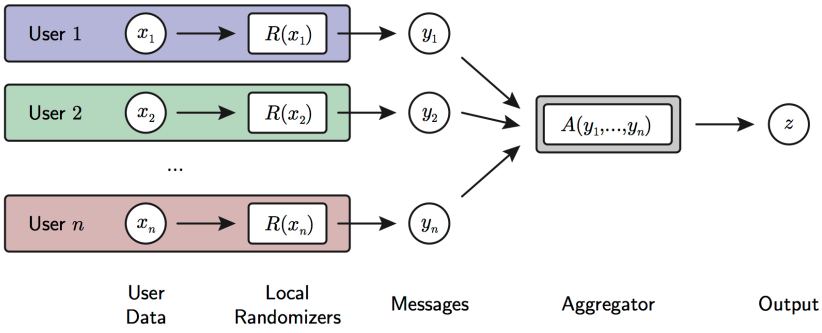

Figure 1: The structure of a (noninteractive) local protocol.

Despite these benefits, local protocols have significant limitations when compared to private algorithms in the *central model*, in which data are collected and processed by a trusted curator. The most discussed limitation is larger error for the same level of privacy (e.g. Dwork et al., 2006b; Kasiviswanathan et al., 2008; Beimel et al., 2011). In this paper, we initiate a systematic study of a different limitation that we show to be equally fundamental:

Locally differentially private protocols are highly vulnerable to manipulation.

While any algorithm can be manipulated by users who lie about their data, we demonstrate that local algorithms can be manipulated to a far greater extent. As the level of privacy or the size of the input domain increase, an adversary who corrupts a vanishing fraction of the users can effectively prevent the protocol from collecting any useful information about the data of the honest users. This result can be interpreted as showing that local differential privacy opens up new, more powerful avenues for *poisoning attacks*—poisoning the private messages can be far more destructive than poisoning the data itself.

Various attackers might be able to exploit this vulnerability to manipulation for nefarious purposes. In particular, if a company is using locally differentially private protocols to collect user data that it then uses to improve its product, then its rivals would have an incentive to exploit these vulnerabilities to gain a competitive edge. If the goal is distribution estimation, our work implies that the rival only needs to corrupt a small fraction of users to highly skew the estimate in statistical distance. Furthermore, we find a setting where estimates can be vulnerable to a *small number* of corruptions.

Prior work had already noted that a *specific* protocol—randomized response (Warner, 1965)—is vulnerable to manipulation (Ambainis et al., 2004; Moran and Naor, 2006). A concurrent and independent work (Cao et al., 2019) gives an empirical study of the effectiveness of natural manipulation attacks against common protocols. In contrast, we show that manipulation is unavoidable for *any* noninteractive local protocol that solves any one of a few basic problems to sufficiently high accuracy, and systematically identify the

optimal degree of manipulation for each problem. These problems capture computing means and histograms, identifying heavy-hitters, and estimating the distribution of users' data. In particular, our work is the first to identify the domain size as a key factor in determining how vulnerable local protocols must be to manipulation. We also give simple experiments validating our theoretical findings. In addition, these experiments show that two protocols that have exactly identical error absent manipulation can nonetheless have dramatically different performance in the presence of manipulation.

Our results suggest caution when deploying locally differentially private protocols: the architecture is *inherently* vulnerable to manipulation. One way to remedy this is to introduce some mechanism that enforces the correctness of users' randomization, such as physical constraints or an interactivity requirement (Moran and Naor, 2006; Ambainis et al., 2004). Our work also reinforces the importance of efficient cryptographic techniques that emulate central-model algorithms in a distributed setting, such as multiparty computation (Dwork et al., 2006a) or shuffling (Bittau et al., 2017; Cheu et al., 2019). Such protocols already have significant accuracy benefits, and our results highlight their much greater resilience to manipulation.

1.1. **Why are Local Protocols Vulnerable to Manipulation?** Intuitively, because local differential privacy requires that each user's message is almost independent of their data, large changes in the users' data induce only small changes in the distribution of the messages. As a result, the aggregator must be highly sensitive to small changes in the distribution of messages. That is, an adversary who can cause small changes in the distribution of messages can make the messages appear as if they came from users with very different data, forcing the aggregator to change its output dramatically.

We can see how this occurs using the classic *randomized response* protocol. Here, each user's has data $x_i \in \{\pm 1\}$ and the objective is to estimate the mean $\frac{1}{n}\sum_{i=1}^n x_i$. For roughly 2ε-local differential privacy, each user outputs

$$y_i = \begin{cases} x_i & \text{with probability } \frac{1+\varepsilon}{2} \\ -x_i & \text{with probability } \frac{1-\varepsilon}{2} \end{cases}$$

so that $\mathbb{E}[y_i] = \varepsilon x_i$. The aggregator computes an unbiased estimate of the mean by returning $\frac{1}{n}\sum_{i=1}^n \frac{y_i}{\varepsilon}$.

In order to extract the relatively weak signal and make the estimate unbiased, the aggregator scales up each message y_i by a factor of $\frac{1}{\varepsilon}$, which increases the influence of each message. This means that an adversary who can flip m of the messages y_i from -1 to $+1$ will increase the aggregator's output by $\frac{2m}{\varepsilon n}$. A simple consequence of our work is that *any* noninteractive LDP protocol for computing the average of bits is similarly vulnerable to manipulation.

1.2. **Frequency Estimation: A Representative Example.** We can more fully illustrate our work results through the example of *frequency estimation*. Consider a protocol whose goal is to collect the frequency of words typed by users on their keyboard. We assume that there are n users, and each user contributes only a single word to the dataset, so each user's word is an element of $[d] = \{1, \ldots, d\}$ where d is the size of the dictionary. The goal of the protocol is to estimate the vector consisting of the frequency of each word as accurately as possible. In this example, we measure accuracy in the ℓ_1 norm (or, equivalently, in statistical

distance or total variation distance): if $v \in \mathbb{R}^d$ is the frequency vector whose entries v_j are the fraction of users whose data takes the value j, and $\hat{v}$ is the estimated frequency vector, then the error is $\|v - \hat{v}\|_1 = \sum_{j=1}^{d} |v_j - \hat{v}_j|$.

Baseline Attacks. In order for the attack to be a concern, the adversary has to be able to introduce more error than what would otherwise exist in the protocol, and the attack should be specific to local differential privacy. In particular, we say the attack is nontrivial if it introduces more error than the following trivial *baselines*:

No Manipulation. The adversary could choose not to manipulate the messages at all, in which case the protocol will still incur some error due to the fact that it must ensure local differential privacy. For example, it is known that an optimal ε-differentially private local protocol for frequency estimation introduces error $\approx \sqrt{d^2/\varepsilon^2 n}$ (Duchi et al., 2013b).

Input Manipulation. The adversary could have the corrupted users change only their inputs. That is, the corrupted users could honestly carry out the protocol as if their data were some arbitrary x_i' instead of x_i (see Figure 2). Since the corrupted users control an m/n fraction of the data, they can skew the overall distribution by m/n. This attack applies to *any* protocol, private or not.

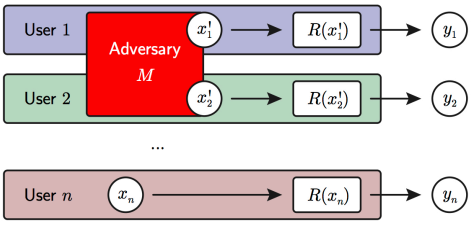

Figure 2: An input-manipulation attack.

These baselines make sense in the context of any task, and we will use the bounds for these baselines to calibrate the effectiveness of attacks for other problems (not just frequency estimation) in the next section.

Our Work: Manipulation Attacks. We consider a general attack model where the adversary is able to corrupt a set of m out of the n users' devices, and can instruct these users to send arbitrary messages, possibly in a coordinated fashion; we visualize this model in Figure 3. The corruptions are unknown to the aggregator running the protocol to prevent the aggregator from ignoring the messages of the corrupted users. In this, and all of our examples, the adversary's goal is to make the error as large as possible—exactly opposite to the goal of the protocol.

In Section 4, we describe and analyze an attack that skews the overall distribution by $\approx \frac{m\sqrt{d}}{\varepsilon n}$, for *any* noninteractive ε-differentially private local protocol. This attack introduces much larger error—by about a $\frac{\sqrt{d}}{\varepsilon}$ factor—than input manipulation, and thus shows specifically that locally private protocols are highly vulnerable to manipulation. We also show our attack is near-optimal by giving a protocol that achieves optimal error in the absence of manipulation and cannot be manipulated by more than $\approx \frac{m\sqrt{d}}{\varepsilon n}$.

For comparison, an adversary of a centrally private algorithm is limited to input manipulation. This is because each user communicates their data noiselessly: in the mean estimation example, the aggregator has no need to increase the influence of each user. Additionally, techniques that simulate centrally private algorithms in a distributed setting such as multiparty computation and shuffling can inherit this resilience.

Measuring the Effectiveness of Attacks. In this work we establish tight upper and lower bounds on the error introduced by manipulation in terms of the parameters n, m, ε, and d. To reduce the number of parameters, and facilitate easier comparisons to the baseline attacks, we have identified two key thresholds that we can use to understand the effectiveness of manipulation attacks for a given task.

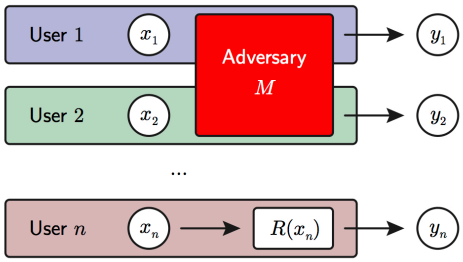

Figure 3: A general manipulation attack.

The first is what we call the *breakdown point*, which is the minimum fraction of users at which the protocol can no longer guarantee non-trivial accuracy. For all problems we consider, the accuracy is non-trivial if it is smaller than some fixed constant (where the choice of constant will not affect the asymptotic bounds). Our attack demonstrates that, for frequency estimation, the breakdown point is roughly $\frac{\varepsilon}{\sqrt{d}}$. That is, that this number of corrupted users can skew the distribution by $\Omega(1)$ in ℓ_1 norm, while *any* two frequency vectors have ℓ_1 distance at most 2. Thus, when ε is small or d is large, an attacker controlling a vanishing fraction of the users can prevent the protocol from achieving any nontrivial accuracy guarantee.

The second threshold is what we call the *significance point*, which is the minimum fraction of users that can increase the error significantly beyond the error necessary to solve the problem absent manipulation. That is, the corrupted users can introduce error on the same order as the error of an optimal protocol with no manipulation. For the frequency estimation problem, the optimal error absent manipulation is $\sqrt{d^2/\varepsilon^2 n}$, and thus the significance point is $\sqrt{d/n}$.

1.3. **Summary of Results: Lower Bounds.** In this work, we construct two manipulation attacks on locally differentially private protocols, and use these attacks to derive lower bounds on the degree of manipulation allowed by local protocols for a variety of tasks (including the frequency estimation example above). We also study the resilience of specific protocols to manipulation. For each problem, we give a protocol that is asymptotically optimal with respect to both ordinary accuracy (i.e., without manipulation) and resilience to manipulation. We also show that popular protocols for most tasks are much less resistant to manipulation than optimal ones.

Below, we first discuss the attacks informally, and then discuss the set of problems to which they apply. We defer details of the attack model to Section 2.2. Our results are summarized in Table 1.

An Attack for Binary Data. Our first attack concerns the simplest problem in local differential privacy—computing a mean of bits. Each user has data $x_i \in \{0, 1\}$, and we assume that each x_i is drawn independently from the Bernoulli distribution $\mathbf{Ber}(p)$, meaning $x_i = 1$ with probability p and $x_i = 0$ with probability $1 - p$. Our goal is to estimate the mean p as accurately as possible. More generally, we could allow the users to have arbitrary data $x_1, \ldots, x_n \in \{0, 1\}$ and try to estimate $\frac{1}{n}\sum_{i=1}^{n} x_i$. For the purposes of attacks, considering the distributional version only makes our results stronger.

Without manipulation, this problem is solved by the classical randomized response protocol (Warner, 1965), which achieves optimal error $\Theta(\frac{1}{\varepsilon\sqrt{n}})$. As we discussed in the introduction, one can show that the error of randomized response increases to $\Theta(\frac{1}{\varepsilon\sqrt{n}} + \frac{m}{\varepsilon n})$ when an adversary corrupts m of the users. We show that no protocol can improve this bound.

Theorem 1.1 (Informal)**.** *For every ε-differentially private local protocol Π for n users with input domain $\{0,1\}$, there is an attack M corrupting m users such that Π cannot distinguish between the following cases:*

(1) *The data is drawn from $\mathbf{Ber}(p_0)$ for $p_0 = \frac{1}{2}$ and Π has been manipulated by M.*
(2) *The data is drawn from $\mathbf{Ber}(p_1)$ for $p_1 = \frac{1}{2} + \Theta(\frac{1}{\varepsilon}(\frac{1}{\sqrt{n}} + \frac{m}{n}))$ and Π has not been manipulated.*

This theorem—combined with existing lower bounds for locally differentially private estimation—shows that, when the data is drawn from $\mathbf{Ber}(p)$ for unknown p, no protocol Π can estimate p and guarantee accuracy better than $\Theta(\frac{1}{\varepsilon\sqrt{n}} + \frac{m}{\varepsilon n})$. As an immediate consequence, when the data $x_1, \ldots, x_n \in \{0,1\}$ may be arbitrary, no protocol Π can estimate the mean $\frac{1}{n}\sum_i x_i$ with significantly better accuracy. Concretely, the $\frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \frac{m}{2n}$ error due to manipulation is within a factor of 4 of the upper bound that can be proved for randomized response. We have not attempted to optimize this constant factor, and we would conjecture that randomized response has exactly optimal robustness to manipulation.

Attacks for Large Domains. Since estimating the mean of bits is a special case of most problems studied in the local model, this attack already shows that manipulation can cause additional error of $\Omega(\frac{m}{\varepsilon n})$ for many problems. In some cases, this bound is already near-optimal, and some protocol achieves a similar upper bound. However, for many cases of interest (such as the frequency estimation example), protocols become more vulnerable to manipulation when the size of the input domain increases. Our second result is an attack on any protocol accepting inputs from the domain $[d] = \{1, \ldots, d\}$ for large d, showing that manipulation can skew the distribution by $\tilde{\Omega}(\frac{m\sqrt{d}}{\varepsilon n})$ without being detected.

Theorem 1.2 (Informal)**.** *For every ε-differentially private local protocol Π for n users with input domain $[d]$, there is an attack M corrupting m users such that Π cannot distinguish between the following cases:*

(1) *The data is drawn from the uniform distribution $\mathbf{U}$ over $[d]$ and M manipulates Π.*
(2) *The data is drawn from some distribution $\mathbf{P}$ with $\|\mathbf{U} - \mathbf{P}\|_1 = \Theta(\frac{1}{\varepsilon}\sqrt{\frac{d}{\log n}}(\frac{1}{\sqrt{n}} + \frac{m}{n}))$ and Π has not been manipulated.*

For a class of natural protocols, the bound on $\|\mathbf{U} - \mathbf{P}\|_1$ can be sharpened to $\Theta(\frac{\sqrt{d}}{\varepsilon}(\frac{1}{\sqrt{n}} + \frac{m}{n}))$.

A consequence of this attack for the example of frequency estimation above is that any local protocol can have the distribution skewed by $\tilde{\Omega}(\frac{m\sqrt{d}}{\varepsilon n})$. As we show in Section 5, this bound is actually matched by a simple protocol. In order to simplify the proof and obtain a statement that applies to arbitrary protocols, we do not optimize the constant factors hidden by the $\Theta(\cdot)$ notation. However, we do give proof-of-concept experiments in Section 7 showing the concrete effect of our attack on a widely studied frequency estimation protocol.

1.4. **Summary of Results: Optimal Protocols.** We consider a variety of tasks of interest in local differential privacy. For each, we show that one of the attacks above gives an optimal bound on the vulnerability of protocols for that task. The results are summarized in Table 1.

Most tasks we consider can be formulated as instances of the following ℓ_p/ℓ_q-*mean estimation* problem for vectors in $\mathbb{R}^d$. [1] Each user's data x_i is a vector in $\mathbb{R}^d$ such that the ℓ_p-norm of each data point is bounded, $\|x_i\|_p \leq 1$. The protocol's goal is to output an estimate of the mean $\hat{\mu}$ with low error in the ℓ_q-norm, $\|\hat{\mu} - \frac{1}{n}\sum_{i=1}^n x_i\|_q$. Recall that $\|v\|_p = (\sum_i v_i^p)^{1/p}$ and $\|v\|_\infty = \max_i |v_i|$. This setup captures a number of widely studied problems:

- The frequency estimation example above is a special case of ℓ_1/ℓ_1 estimation, where each user represents their word $x_i \in [d]$ by the standard basis vector $e_{x_i} \in \mathbb{R}^d$ with a 1 in the x_i-th coordinate and 0 elsewhere.
- Computing a histogram of data in $[d]$ is a special case of ℓ_1/ℓ_∞-mean estimation. The *heavy-hitters (HH)* problem, which asks one only to identify the heaviest bins of a histogram and their frequencies, suffices to solve ℓ_1/ℓ_∞-mean estimation, so manipulation attacks on the latter thus imply attacks on the former. Computing heavy-hitters has been a focus of the past few years (Hsu et al., 2012; Bassily and Smith, 2015; Bassily et al., 2017; Bun et al., 2018), and it is central to systems deployed by Google (Erlingsson et al., 2014) and Apple (Apple Differential Privacy Team, 2017).
- Computing the answers to d statistical queries (Kearns, 1993; Blum et al., 2005; Kasiviswanathan et al., 2008) is a special case of ℓ_∞/ℓ_∞-mean estimation. Users have data in some arbitrary domain $\mathcal{X}$, there are d query functions $f_1, \ldots, f_d : \mathcal{X} \to [-1, 1]$, and we would like an accurate estimate of $\sum_{i=1}^n f_j(x_i)$ for every j. In the corresponding mean estimation instance, $x_i = (f_1(x_i), \ldots, f_d(x_i))$.
- When minimizing a sum of convex functions $f(\theta) = \sum_{i=1}^n f_{x_i}(\theta)$ defined by the users' data (e.g. to train a machine learning model), one often computes the average gradient $\sum_{i=1}^n \nabla f_{x_i}(\theta_t)$ at a sequence of points θ_t. Typically one assumes that the gradients are bounded in ℓ_2, and convergence requires an accurate estimate in ℓ_2, making this an instance of ℓ_2/ℓ_2-mean estimation. (More generally, optimization *requires* this sort of estimation (Bassily et al., 2014)).
- We study one further problem, ℓ_1/ℓ_1-*uniformity testing*, for which Acharya et al. (2019) gave optimal LDP protocols. Assuming the data is drawn from some distribution over $[d]$, we want to determine if this distribution is either uniform or is far from uniform in ℓ_1 distance.

Since every ℓ_p/ℓ_q mean estimation problem generalizes binary mean estimation (the special case where $d = 1$), our first attack gives a lower bound on all of these problems. Our second attack is precisely an attack on the ℓ_1/ℓ_1-testing problem, and thus implies a lower bound of $\tilde{\Omega}(\frac{m\sqrt{d}}{\varepsilon n})$ for that problem. Finally, since the ℓ_1/ℓ_1-mean estimation problem strictly generalizes ℓ_1/ℓ_1-testing problem—once we estimate the mean, we can determine if it is close to uniform or far from uniform—we obtain the same lower bound for that problem.

Resilient Protocols. While all of our optimal protocols were known prior to our work, we demonstrate that the choice of protocol is crucial. Some well known protocols with optimal accuracy absent manipulation allow for much greater manipulation than necessary. For

[1] Given any vector $v \in \mathbb{R}^d$ and any $p \geq 1$, the ℓ_p-norm is defined as $\|x\|_p = (\sum_{j=1}^d |x_j|^p)^{1/p}$. For $p = \infty$, the ℓ_∞ norm is defined as $\max_{j=1,\ldots,d} |x_j|$.

Problem	No Manipulation	Manipulation	Breakdown Pt.	Significance Pt.
ℓ_1/ℓ_1 Estimation (Freq. Estimation)	$\Theta\left(\sqrt{\frac{d^2}{\varepsilon^2 n}}\right)$ (Duchi et al., 2013b)	$\tilde{O}(\frac{m}{n}\cdot\frac{\sqrt{d}}{\varepsilon})$ Thm 5.7 $\Omega(\frac{m}{n}\cdot\frac{\sqrt{d}}{\varepsilon\sqrt{\log n}})$ Thm 4.10 *	$O\left(\varepsilon\sqrt{\frac{\log n}{d}}\right)$	$O\left(\sqrt{\frac{d\log n}{n}}\right)$
ℓ_1/ℓ_1 Testing (Uniformity Testing)	$\Theta\left(\sqrt{\frac{d}{\varepsilon^2 n}}\right)$ (Acharya et al., 2019)	$O(\frac{m}{n}\cdot\frac{\sqrt{d}}{\varepsilon})$ Thm 5.9 $\Omega(\frac{m}{n}\cdot\frac{\sqrt{d}}{\varepsilon\sqrt{\log n}})$ Thm 4.9 *	$O\left(\varepsilon\sqrt{\frac{\log n}{d}}\right)$	$O\left(\sqrt{\frac{\log n}{n}}\right)$
ℓ_1/ℓ_∞ Estimation (Histograms / HH)	$\Theta\left(\sqrt{\frac{\log d}{\varepsilon^2 n}}\right)$ (Bassily and Smith, 2015)	$O(\frac{m}{n}\cdot\frac{1}{\varepsilon})$ Thm 5.4 $\Omega(\frac{m}{n}\cdot\frac{1}{\varepsilon})$ Thm 3.4	$O(\varepsilon)$	$O\left(\sqrt{\frac{\log d}{n}}\right)$
ℓ_∞/ℓ_∞ Estimation (d statistical queries)	$\Theta\left(\sqrt{\frac{d\log d}{\varepsilon^2 n}}\right)$ [Folklore]	$O(\frac{m}{n}\cdot\frac{1}{\varepsilon})$ Thm 5.2 ♯ $\Omega(\frac{m}{n}\cdot\frac{1}{\varepsilon})$ Thm 3.4	$O(\varepsilon)$	$O\left(\sqrt{\frac{d\log d}{n}}\right)$
ℓ_2/ℓ_2 Estimation (Gradients)	$\Theta\left(\sqrt{\frac{d}{\varepsilon^2 n}}\right)$ (Duchi et al., 2013a)	$\tilde{O}(\frac{m}{n}\cdot\frac{1}{\varepsilon})$ Thm 5.8 $\Omega(\frac{m}{n}\cdot\frac{1}{\varepsilon})$ Thm 3.4	$O(\varepsilon)$	$O\left(\sqrt{\frac{d}{n}}\right)$

Table 1: Summary of Results, assuming $\varepsilon = O(1)$. For each problem, we present existing results for the optimal error under local privacy without manipulation. We also list our upper and lower bounds on the error from manipulation attacks. ♯ indicates an upper bound limited to public-string-oblivious attacks and * indicates that a $\sqrt{\log n}$ factor can be removed for a natural class of protocols. In each case, no protocol can guarantee nontrivial accuracy in the presence of [Breakdown Point] corrupted users. When there are [Significance Point] corrupted users, the error they introduce eclipses the error absent manipulation.

example, the simplest adaptation of randomized response to frequency estimation, in which each player sends one bit per potential item, allows m corrupted users to introduce error about $md/\epsilon n$ in a direction of their choice, which is about $\sqrt{d}$ larger than optimal.

1.5. **Overview of Techniques.** *Attack for Binary Data.* Our argument boils down to proving the following claim: for every ε-differentially private local protocol, there is some attacker who corrupts each user independently with probability $\frac{m}{n}$ in such a way that data drawn from the uniform distribution over $\{\pm 1\}$ appears to have mean $\approx \frac{m}{\varepsilon n}$. To show this, we derive the following from (Kairouz et al., 2015): for any ε-differentially private local randomizer R and distribution $\mathbf{Rad}(\mu)$ over $\{\pm 1\}$ with mean μ, the distribution $R(\mathbf{Rad}(\mu))$

is exactly a mixture $R^{(\mu)}$ of two distributions R^+ and R^- where

$$R^{(\mu)} \approx \tfrac{1+\varepsilon\mu}{2} \cdot R^+ + \tfrac{1-\varepsilon\mu}{2} \cdot R^-.$$

Since the data and messages are independent and identically distributed (iid), the messages consist of n iid samples from $R^{(\mu)}$. If $\mu = 0$, but an attacker corrupts each user independently with probability $\frac{m}{n}$, and has the corrupted users send a message sampled from R^+, then the messages remain independent and consist of n messages sampled from $R^{(\mu)}$ for $\mu = \frac{m}{\varepsilon n}$, exactly the same if there were no corruptions but $\mu = \frac{m}{\varepsilon n}$. Since the aggregator cannot distinguish these two identical distributions, it must have error at least $\approx \frac{m}{2\varepsilon n}$ on one of them. Some technicalities arise in the proof because (1) the attacker has a fixed budget of m corruptions that might be exceeded when corrupting each user independently, and (2) the local randomizer might only satisfy (ε, δ)-differential privacy, and thus might have a slightly more complex structure.

Attack for High-Dimensional Data. For high-dimensional data, we can show the existence of a distribution R^+ that has an even more extreme effect on the overall distribution of messages than in the binary case. For any distribution S on the domain $\{1, \ldots, d\}$, let $R^{(S)}$ be the distribution on messages $R(S)$. Let U be the uniform distribution on the domain. We show (roughly) that for every ε-differentially private local randomizer R, there is some distribution S supported on $d/2$ domain elements, such that $R^{(U)}$ and $R^{(S)}$ are only $\varepsilon/\sqrt{d}$ apart, and there exists an extreme distribution $R^+ \approx R^{(U)} + \frac{1}{\varepsilon}(R^{(S)} - R^{(U)})$. Thus, if we corrupt only about an $\varepsilon/\sqrt{d}$ fraction of users, we can make messages from $R^{(U)}$ look like messages from $R^{(S)}$. Since S and U have distance at least $1/2$, corrupting about an $\varepsilon/\sqrt{d}$ fraction of users is enough to make the error at least $1/2$. With some rescaling we can prove the bound that we claim for an arbitrary number of corruptions. For technical reasons, our formal proof works by a reduction to the binary attack, in which we argue that any ε-differentially private protocol for frequency estimation can be used to get a protocol for binary estimation that is $\approx (\varepsilon/\sqrt{d})$-differentially private.

Optimally Robust Protocols. All of the optimally robust protocols we present have already appeared in the literature, but had not been analyzed with respect to manipulation attacks. However, not all protocols with optimal accuracy without manipulation have optimal robustness to manipulation. In particular, the protocols that we show are optimally robust use the public-coin model to reduce the amount of communication per user down to a single bit (see e.g. Kasiviswanathan et al., 2008; Bassily and Smith, 2015), and thereby dramatically decreases the space of possible manipulation.

1.6. **Related Work.** *Manipulation Attacks.* Prior work had already observed that the specific randomized response protocol was vulnerable to manipulation (Ambainis et al., 2004; Moran and Naor, 2006). In contrast to ours, these works constructed efficient cryptographic protocols for sampling from the correct distribution, which resist our attacks. Our work shows that some degree of cryptography is necessary to avoid manipulation. A concurrent and independent work (Cao et al., 2019) performed an empirical study of simple manipulation attacks on common protocols for tasks like frequency estimation and heavy-hitters. In contrast to ours, their work does not prove any inherent limitations on the robustness of local protocols to manipulation, nor does it establish the crucial role that the domain size plays.

Our work is loosely related to *data poisoning attacks* in adversarial machine learning. In data poisoning, the adversary is inserts additional data to somehow degrade the quality of the output. Our attacks can be viewed as data poisoning attacks where the "data" being poisoned is actually the messages to the protocol. Thus, our results can be viewed as showing that adding local randomization to achieve privacy makes the protocol much more vulnerable to data poisoning.

Our work is also related to the literature on *robust statistics*. In the standard model of robust statistics, we are given data drawn from distribution $\mathbf{P}$ with some structure (e.g. $\mathbf{P}$ is a Gaussian distribution), but some small fraction of the data has been corrupted with arbitrary data, and the goal is to identify the distribution $\mathbf{P}$ as well as possible. Our setting is similar except that we don't get access to the data directly, but only once its been filtered through some set of private local randomizers. One might hope to obtain local protocols that are robust to manipulation using techniques from robust statistical estimators on the distribution of messages induced by the local randomizers. Our attacks can be viewed as showing that such robust estimators don't exist.

Cryptographic Approaches. Our work reinforces the importance of efficient cryptographic techniques that emulate central-model algorithms in a distributed setting. Multiparty computation (MPC) allows a network of parties to jointly execute a randomized algorithm on encrypted or secret-shared data while exposing only the final result of the computation. The value of simulating a differentially private computation was first highlighted in (Dwork et al., 2006a; Beimel et al., 2011). Briefly, the MPC approach gets the accuracy of the central model, and limits attackers to input manipulation, which is unavoidable without some outside certification of inputs. The downside of this approach is computational efficiency. Despite recent advances in practical MPC, applications like collecting information about mobile data usage place extreme demands on protocols that make current solutions difficult to use. To our knowledge, known MPC protocols either scale poorly to large networks, assume an honest-but-curious server (e.g. Bonawitz et al., 2017), or leak extra, hard-to-reason-about intermediate results from a computation. Although the MPC literature is too vast to survey here, we refer the reader to a recent survey of the issues that arise in *federated learning* for a (Kairouz et al., 2019) more thorough discussion of these issues.

One recent approach asks whether we can reduce important differentialy private algorithms to some simple primitive which is easier to implement in MPC. For example, the *shuffled model* (Bittau et al., 2017; Cheu et al., 2019; Erlingsson et al., 2019) assumes the availability of a trusted shuffling primitive, which anonymizes the origin of the messages by applying a secret permutation before delivering them to the aggregator. That model allows accuracy close to that of the central model for several tasks but leaves open just how well the shuffler can be implemented by a real protocol. On the other hand, shuffled protocols for histograms are more resilient than counterparts in the local model. Cheu et al. (2019), for example, give a protocol where the influence of each message is scaled by a factor close to 1 instead of $\frac{1}{\varepsilon}$ as in the local model.

Finally, cryptographic protocols can be used in a much narrower and potentially scalable way to ensure that local-model protocols are carried out without manipulation (see Ambainis et al., 2004, for a protocol tailored to binary randomized response). These require some interaction between clients and the server and retain the accuracy limitations of the local model, but can constrain the client to simple input manipulation. Specific physical devices, such as carefully generated scratch cards, can also provide such a guarantee (Moran and

Naor, 2006). Current techniques for efficient MPC should suffice for wider use of such protocols.

1.7. **Organization.** In Section 2 we introduce the model and key concepts. In Section 3, we demonstrate attacks on protocols for binary data, and in Section 4, we demonstrate attacks on protocols for large data domains. In Section 5 we identify protocols with near-optimal resistance to manipulation for a variety of canonical problems in local differential privacy. In Section 6 we highlight the fact that not all protocols with optimal error absent manipulation are optimally robust to manipulation.

2. Threat Model and Preliminaries

2.1. **Local Differential Privacy.** In this model there are n *users*, and each user $i \in [n]$ holds some sensitive data $x_i \in \mathcal{X}$ belonging to some *data universe* $\mathcal{X}$. There is also a public random string S. Finally there is a single *aggregator* who would like to compute some function of the users' data $x_1, \ldots, x_n$. In this work, for simplicity, we restrict attention to *non-interactive local differential privacy*, meaning the users and the aggregator engage in the following type of protocol:

(1) A public random string S is chosen from some distribution $\mathbf{S}$ over support $\mathcal{S}$.
(2) Each user computes a *message* $y_i \leftarrow R_i(x_i, b)$ using a *local randomizer* $R_i : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$.
(3) The aggregator $A : \mathcal{Y}^n \times \mathcal{S} \to \mathcal{Z}$ computes some output $z \leftarrow A(y_1, \ldots, y_n, S)$.

Thus the protocol Π consists of the tuple $\Pi = ((R_1, \ldots, R_n), A, \mathbf{S})$. We will sometimes write $\vec{R}$ to denote the local randomizers $(R_1, \ldots, R_n)$. If $R_1 = \cdots = R_n = R$ then we say the protocol is *symmetric* and denote it $\Pi = (R, A, \mathbf{S})$.

Given user data $\vec{x} \in \mathcal{X}^n$ we will write $\Pi(\vec{x})$ to denote the distribution of the protocol's output when the users' data is $\vec{x}$, and $\vec{R}(\vec{x})$ denotes the distribution of the protocol's messages. Given a distribution $\mathbf{P}$ over $\mathcal{X}$, we will write $\Pi(\mathbf{P})$ and $\vec{R}(\mathbf{P})$ to denote the resulting distributions when $\vec{x}$ consists of n independent samples from $\mathbf{P}$.

Informally, we say that the protocol satisfies *local differential privacy* (Evfimievski et al., 2003; Dwork et al., 2006b; Kasiviswanathan et al., 2008) if the local randomizers depend only very weakly on their inputs. Formally,

Definition 2.1 (Local DP (Evfimievski et al., 2003; Dwork et al., 2006b; Kasiviswanathan et al., 2008))**.** A protocol $\Pi = ((R_1, \ldots, R_n), A, \mathbf{S})$ satisfies (ε, δ)-*local differential privacy* if for every $i \in [n]$, every $x, x' \in \mathcal{X}$, every $S \in \mathcal{S}$ and every $Y \subseteq \mathcal{Y}$,

$$\mathop{\mathbb{P}}_{R_i}[R_i(x, S) \in Y] \leq e^{\varepsilon} \cdot \mathop{\mathbb{P}}_{R_i}\big[R_i(x', S) \in Y\big] + \delta$$

where we stress that the randomness is *only* over the coins of R_i. If $\delta = 0$, we simply write ε-*local differential privacy*.

2.2. **Threat Model: Manipulation Attacks.** We capture manipulation attacks via a game involving a protocol $\Pi = (\vec{R}, A, \mathbf{S})$, a vector $\vec{x}$ of n data values, and an adversary M. We parameterize the game by the number of users n and the number of corrupted users $m \leq n$, written as $\text{Manip}_{m,n}$; when clear from context, the subscript is omitted. The crux of the game is that the adversary corrupts a set C of at most m users: a member of that set *manipulates* by playing some arbitrary message chosen by the adversary. Meanwhile, any other user i plays *honestly* by sending the message $y_i = R_i(x_i, S)$. Figure 3 presents the structure of an attack in the case where $C = \{1, 2\}$.

The game is described in Figure 4, including a possible restriction on the attacker. We use $\text{Manip}_{m,n}(\Pi, \vec{x}, M)$ to denote the distribution on outputs of the protocol on data $\vec{x}$ and messages manipulated by M, and $\text{Manip}_{m,n}(\vec{R}, \vec{x}, M)$ to denote the distribution of messages in the protocol. Given a distribution $\mathbf{P}$ over $\mathcal{X}$, we will use $\text{Manip}_{m,n}(\Pi, \mathbf{P}, M)$ and $\text{Manip}_{m,n}(\vec{R}, \mathbf{P}, M)$ to denote the resulting distributions when $\vec{x}$ consists of n independent samples from $\mathbf{P}$.

Parameters: $0 \leq m \leq n$.
Elements: A protocol $\Pi = (\vec{R}, A, \mathbf{S})$ for n users, a vector of data $\vec{x}$, an attacker M.

(1) Each user i is given data x_i.
(2) The public string $S \sim \mathbf{S}$ is sampled.
(3) The attacker M chooses a set of *corrupted users* $C \subseteq [n]$ of size $\leq m$.
    If the corruptions are independent of the public string S then they are *public-string-oblivious*, and otherwise they are *public-string-adaptive*.
(4) The attacker M chooses a set of messages $\{y_i\}_{i \in C}$ for the corrupted users.
(5) The non-corrupted users $i \notin C$ choose messages $y_i \sim R_i(x_i, S)$ honestly.
(6) The aggregator returns $z \leftarrow A(y_1, \ldots, y_n, S)$.

Figure 4: Manipulation Game $\text{Manip}_{m,n}$

2.3. **Notational Conventions.** Throughout, boldface roman letters indicate distributions (e.g. $\mathbf{P}$). Vectors are denoted $\vec{v} = (v_1, v_2, \ldots)$. We write $[n]$ to denote the set $\{1, \ldots, n\}$.

We use $\mathbf{Rad}(\mu)$ to denote the distribution over $\{\pm 1\}$ with mean μ, so $\mathbb{P}[\mathbf{Rad}(\mu) = +1] = \frac{1+\mu}{2}$. Note that $\mathbf{Rad}(0)$ is uniform on $\{\pm 1\}$.

3. Attacks Against Protocols for Binary Data

In this section, we show how to attack any protocol that estimates the mean of a Rademacher distribution $\mathbf{Rad}(\mu)$. [2] In particular, we show that any such protocol has error $\Omega(\frac{m}{\varepsilon n})$ in the presence of m corrupt users.

We begin with the result from (Kairouz et al., 2015) that decomposes any differentially private randomizer into a mixture of distributions:

[2]The choice of data universe $\mathcal{X} = \{\pm 1\}$ simplifies the analysis but is not inherent to the results; any binary data universe has corresponding attacks.

Lemma 3.1 (Adapted from Kairouz et al. (2015))**.** *If $R : \{\pm 1\} \to \mathcal{Y}$ satisfies (ε, δ)-differential privacy, then there exist distributions $R^{(+1)}, R^{(-1)}, R^{\perp}, R^{\top}$ such that $R(+1)$ and $R(-1)$ are mixtures between them:*

$$R(+1) = \frac{e^{\varepsilon}}{e^{\varepsilon}+1} \cdot (1-\delta) \cdot R^{(+1)} + \frac{1}{e^{\varepsilon}+1} \cdot (1-\delta) \cdot R^{(-1)} + \delta \cdot R^{\perp}$$
$$R(-1) = \frac{1}{e^{\varepsilon}+1} \cdot (1-\delta) \cdot R^{(+1)} + \frac{e^{\varepsilon}}{e^{\varepsilon}+1} \cdot (1-\delta) \cdot R^{(-1)} + \delta \cdot R^{\top}$$

The analysis of our attack will assume data is drawn from a distribution, so the following corollary will be useful:

Corollary 3.2. *If $R : \{\pm 1\} \to \mathcal{Y}$ satisfies (ε, δ)-differential privacy, then there exist distributions $R^{(+1)}, R^{(-1)}$ such that, for all $\mu \in [-1, +1]$, $R(\mathbf{Rad}(\mu))$ is within statistical distance δ of the mixture $(\frac{1}{2} + \frac{e^{\varepsilon}-1}{e^{\varepsilon}+1} \cdot \frac{\mu}{2}) \cdot R^{(+1)} + (\frac{1}{2} - \frac{e^{\varepsilon}-1}{e^{\varepsilon}+1} \cdot \frac{\mu}{2}) \cdot R^{(-1)}$.*

Our attack, Algorithm 1 below, takes advantage of this structure of R by skewing the mixture ratio. Hence, no aggregator can tell if messages were generated from data with large mean or by manipulating the protocol.

Algorithm 1: A manipulation attack $M^{\vec{R}}_{m,n}$ against any protocol using n differentially private randomizers $\vec{R}$

For each $i \in [n]$:
(1) Add i to C with probability $m/2n$.
(2) If $|C| = m$ break the loop

For each corrupted user $i \in C$, report $y_i \sim R_i^{(+1)}$.

Lemma 3.3. *For any $n > m > 18$ and any n randomizers $\vec{R}$ that satisfy (ε, δ)-differential privacy, the distribution $\mathrm{Manip}_{m,n}(\vec{R}, \mathbf{Rad}(0), M^{\vec{R}}_{m,n})$ cannot be distinguished from $\vec{R}(\mathbf{Rad}(\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1} \cdot \frac{m}{2n}))$ with arbitrarily low probability of failure. Specifically, the statistical distance is at most $1/10 + 2n\delta$.*

Proof. In the first part of the proof, we will argue that $M^{\vec{R}}_{m,n}$ behaves similarly to the alternative attack $\widetilde{M}^{\vec{R}}_{m,n}$ in which we eliminate step (2) of the for loop and choose whether or not to corrupt each user independently. Note that this attack will not always satisfy our budget of m corruptions, so it is not a valid attack in our model, but it is nonetheless useful for the analysis. The second part shows that $\mathrm{Manip}_{m,n}(\vec{R}, \mathbf{Rad}(0), \widetilde{M}^{\vec{R}}_{m,n})$ is approximately the same as having each user i independently sample from the mixture

$$\mathbf{P}_i := (\tfrac{1}{2} + \tfrac{m}{4n}) R_i^{(+1)} + (\tfrac{1}{2} - \tfrac{m}{4n}) R_i^{(-1)}.$$

The final part invokes Corollary 3.2 to approximate $\vec{\mathbf{P}}$ by $\vec{R}(\mathbf{Rad}(\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1} \cdot \frac{m}{2n}))$.

First, we claim that the statistical distance between

$$\mathrm{Manip}_{m,n}(\vec{R}, \mathbf{Rad}(0), M^{\vec{R}}_{m,n})$$

and

$$\mathrm{Manip}_{m,n}(\vec{R}, \mathbf{Rad}(0), \widetilde{M}^{\vec{R}}_{m,n})$$

is at most $1/10$. These distributions only differ in the event that we hit $|C| = m$ and stop the loop early. This happens with probability exactly $\mathbb{P}[\mathbf{Bin}(n, m/2n) > m]$, and by standard bounds, this probability is at most $1/10$ whenever $m \geq 18$.

Next, we argue that the statistical distance between $\text{Manip}_{m,n}(\vec{R}, \mathbf{Rad}(0), \widetilde{M}^{\vec{R}}_{m,n})$ and $\vec{\mathbf{P}}$ is at most $n\delta$. This is achieved by proving that the i-th user's message is sampled from a distribution within δ of $\mathbf{P}_i$. Note that

$$\mathbf{P}_i = \tfrac{m}{2n} \cdot R_i^{(+1)} + (1 - \tfrac{m}{2n})(\tfrac{1}{2} \cdot R_i^{(+1)} + \tfrac{1}{2} \cdot R_i^{(-1)}) \tag{3.1}$$

Corruption status in $\widetilde{M}^{\vec{R}}_{m,n}$ is determined by a Bernoulli process with probability $\frac{m}{2n}$. If corrupted, user i will sample from $R_i^{(+1)}$; this corresponds to first term of (3.1). If not, Corollary 3.2 implies that their message distribution $R_i(\mathbf{Rad}(0))$ is within δ of $\frac{1}{2} \cdot R_i^{(+1)} + \frac{1}{2} \cdot R_i^{(-1)}$; this corresponds to the second term of (3.1). Thus, the i-th distribution is within δ of (3.1). Since each of the n messages are independent, the overall difference between the distributions is at most $n\delta$.

Finally, Corollary 3.2 implies $\vec{R}(\mathbf{Rad}(\frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \frac{m}{2n}))$ is within statistical distance $n\delta$ of $\vec{\mathbf{P}}$. □

A consequence of Lemma 3.3 is that no private protocol can estimate both $\mathbf{Rad}(0)$ and $\mathbf{Rad}(\frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \frac{m}{2n})$ with high accuracy under manipulation.

Theorem 3.4. *For any $n > m > 18$ and any $\delta < 1/20n$, if $\Pi = (\vec{R}, A)$ is an (ε, δ)-differentially private local protocol for n users and with probability $\geq 95/100$ it estimates $\mathbf{Rad}(\frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \frac{m}{2n})$ to within $\frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \frac{m}{4n}$, then with probability $\geq 3/4$ it does not estimate $\mathbf{Rad}(0)$ to within $\frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \frac{m}{4n}$ under attack $M^{\vec{R}}_{m,n}$.*

4. Attacks Against Protocols for Large Data Universes

In this section, we show that more powerful manipulation attacks are possible when the data universe is $[d]$ for $d > 2$. For binary data, our attack showed that for any protocol there are two distributions $\mathbf{U}$ and $\mathbf{P}$ (i.e. $\mathbf{Rad}(0)$ and $\mathbf{Rad}(\mu(m, n, \varepsilon))$ with large statistical distance that are indistinguishable under manipulation. Specifically, $\|\mathbf{U} - \mathbf{P}\|_1 = \Omega(\frac{1}{\varepsilon\sqrt{n}} + \frac{m}{\varepsilon n})$ where $\|\mathbf{U} - \mathbf{P}\|_1$ denotes the ℓ_1 distance between the distributions $\sum_{j=1}^{d} |\mathbf{U}(j) - \mathbf{P}(j)|$. Here, we show that there is an attack and a distribution $\mathbf{P}$ such that $\|\mathbf{U} - \mathbf{P}\|_1 = \Omega\big(\sqrt{\frac{d}{\log n}}(\frac{1}{\varepsilon\sqrt{n}} + \frac{m}{\varepsilon n})\big)$ yet $\mathbf{U}, \mathbf{P}$ are indistinguishable under this attack. This construction implies lower bounds for uniformity testing (given samples from $\mathbf{P}$, determine if $\mathbf{P} = \mathbf{U}$ or if $\|\mathbf{P} - \mathbf{U}\|_1$ is large) and ℓ_1 estimation (given samples from $\mathbf{P}$, report $\mathbf{P}'$ such that $\|\mathbf{P} - \mathbf{P}'\|_1$ is small).

We remark that our analysis will focus on protocols that satisfy pure differential privacy. This is essentially without loss of generality: any powerful attack against pure local privacy implies a powerful attack against approximate local privacy. Our notion of "powerful" is defined below.

Definition 4.1 (Powerful Attacks)**.** Fix any $\alpha : \mathbb{R} \to (0, 1)$ and $\gamma \in (0, 1)$. A manipulation attack M is (α, γ)-*powerful against* (ε, δ)-*locally private protocols* if, for any such protocol $\Pi = (\vec{R}, A)$, there is a distribution $\mathbf{P}$ over $[d]$ where $d_{SD}(\Pi(\mathbf{P}), \text{Manip}_{m,n}(\Pi, \mathbf{U}, M)) < \gamma$, even though $\|\mathbf{P} - \mathbf{U}\|_1 \geq \alpha(\varepsilon)$.

Claim 4.2. Suppose manipulation attack M is (α, γ)-powerful against $(\varepsilon, 0)$-locally private protocols. Then there is an M_δ that is $(\alpha_\delta, \gamma_\delta)$-powerful against (ε, δ)-locally private protocols, where $\alpha_\delta(\varepsilon) := \alpha(2\varepsilon)$ and $\gamma_\delta := \gamma + 2n\delta$.

We prove the above in Appendix A.1. Because it is standard for $\delta = o(1/n)$, we may launch the attack for pure differential privacy and only experience a constant-factor change in the γ parameter.

4.1. **A Family of Data Distributions.** In this section, we show a particular way to convert a Rademacher distribution into a distribution over $[d]$. For a given partition of $[d]$ into $H, \overline{H}$ where $|H| = d/2$, we map the value $+1$ to a uniform element of H and -1 to a uniform element of $\overline{H}$. Thus, when $x \sim \mathbf{Rad}(\mu)$, we obtain a corresponding random variable $\hat{x}$ over $[d]$ whose distribution is $\mathbf{P}_{H,\mu}$ (see (4.1) below). Notice that estimating $\mathbb{P}[\hat{x} \in H]$ implies estimating μ.

$$\mathbf{P}_{H,\mu} := \begin{cases} \text{Uniform over } H \text{ with probability } \frac{1}{2} + \frac{\mu}{2} \\ \text{Uniform over } \overline{H} \text{ otherwise} \end{cases} \tag{4.1}$$

The algorithm $Q_{H,R}$ (Algorithm 2) performs the encoding of binary data $x \in \{\pm 1\}$ into $\hat{x} \in [d]$ then executes the randomizer R. Claim 4.3 is immediate from the construction.

Algorithm 2: $Q_{H,R}$ a local randomizer for binary data

Parameters: *A subset $H \subset [d]$ with size $d/2$; a local randomizer $R : [d] \to \mathcal{Y}$*
Input: $x \in \{\pm 1\}$
Output: $y \in \mathcal{Y}$

If $x = 1$ then sample $\hat{x}$ uniformly from H
Otherwise, sample $\hat{x}$ uniformly from $\overline{H}$.
Return $y \sim R(\hat{x})$

Claim 4.3. For any local randomizer $R : [d] \to \mathcal{Y}$, $H \subset [d]$ with size $d/2$, and $\mu \in [-1, +1]$, the execution of $Q_{H,R}$ (Algorithm 2) on a value drawn from $\mathbf{Rad}(\mu)$ is equivalent to the execution of R on a value drawn from $\mathbf{P}_{H,\mu}$:

$$Q_{H,R}(\mathbf{Rad}(\mu)) = R(\mathbf{P}_{H,\mu})$$

Given n randomizers $\vec{R} = (R_1, \ldots, R_n)$, let $\vec{Q}_H$ denote the vector $(Q_{H,R_1}, \ldots, Q_{H,R_n})$. We can immediately generalize Claim 4.3 to multiple randomizers:

Claim 4.4. For any n randomizers $\vec{R} = (R_1, \ldots, R_n)$ for data universe $[d]$, any $H \subset [d]$ with size $d/2$, and $\mu \in [-1, +1]$, the execution of $\vec{Q}_H$ on a sample from $\mathbf{Rad}(\mu)$ is equivalent with the execution of R on a sample from $\mathbf{P}_{H,\mu}$:

$$\vec{Q}_H(\mathbf{Rad}(\mu)) = \vec{R}(\mathbf{P}_{H,\mu})$$

4.2. **The Attack.** In this subsection, we describe how to attack any differentially private protocol for d-ary data; to remove ambiguity with $M^{\vec{R}}_{m,n}$ (Algorithm 1), the attack will be denoted $M^{\vec{R}}_{d,m,n}$. As specified in Algorithm 3, the first step is to sample a uniformly random H. We show that if all $Q_{H,R_1}, \ldots, Q_{H,R_n}$ satisfy (ε, δ) differential privacy, then this attack inherits guarantees from the previous section. Then we show that the randomizers have strong privacy parameters with constant probability.

We begin the analysis of $M^{\vec{R}}_{d,m,n}$ by considering its behavior *conditioned on a fixed choice of* H. This restricted form will be denoted $M^{\vec{R},H}_{d,m,n}$. Then we analyze how the random choice of H gives the desired lower bound.

Algorithm 3: An attack $M^{\vec{R}}_{d,m,n}$ against any protocol using n differentially private randomizers $\vec{R}$ for d-ary data

Sample H uniformly from all subsets of $[d]$ with size $d/2$
For $i \in [n]$, add i to C with probability $m/2n$.
If $|C| > m$, remove uniformly random members until $|C| = m$.
For each corrupted user $i \in C$, report $y_i \sim Q^{(+1)}_{H,R_i}$

4.2.1. *Analysis for fixed set* H. First, we show that manipulating $\vec{R}$ with $M^{\vec{R},H}_{d,m,n}$ induces the same distribution as if we had manipulated $\vec{Q}_H$ with $M^{\vec{Q}_H}_{m,n}$:

Claim 4.5. Fix any n randomizers $\vec{R} = (R_1, \ldots, R_n)$ for data universe $[d]$, any $m \leq n$, and any $H \subset [d]$ with size $d/2$. If each Q_{H,R_i} is (ε, δ)-differentially private, then for any value $\mu \in [-1, +1]$, the distribution $\text{Manip}(\vec{R}, \mathbf{P}_{H,\mu}, M^{\vec{R},H}_{d,m,n})$ is identical to $\text{Manip}\Big(\vec{Q}_H, \mathbf{Rad}(\mu), M^{\vec{Q}_H}_{m,n}\Big)$

Proof. To simplify the presentation, we assume that the users are sorted so that $C = \{1, \ldots, |C|\}$.

$$
\begin{aligned}
&\text{Manip}\Big(\vec{R}, \mathbf{P}_{H,\mu}, M^{\vec{R},H}_{d,m,n}\Big) \\
&= (Q^{(+1)}_{H,R_i})_{i \leq |C|} \times (R_i(\mathbf{P}_{H,\mu}))_{i > |C|} && \text{(By construction)} \\
&= (Q^{(+1)}_{H,R_i})_{i \leq |C|} \times (Q_{H,R_i}(\mathbf{Rad}(\mu)))_{i > |C|} && \text{(Claim 4.3)} \\
&= \text{Manip}\Big(\vec{Q}_H, \mathbf{Rad}(\mu), M^{\vec{Q}_H}_{m,n}\Big)
\end{aligned}
$$

The final equality follows from the fact that $|C|$ in $M^{\vec{R},H}_{d,m,n}$ is distributed identically with its counterpart in $M^{\vec{Q}_H}_{m,n}$. □

Claims 4.4 and 4.5 imply that we can use the analysis of $M^{\vec{R}}_{m,n}$ to show that our new attack $M^{\vec{R},H}_{d,m,n}$ is powerful, *provided that* (ε, δ)-privacy holds for all Q_{H,R_i}.

Lemma 4.6. *Fix any n randomizers $\vec{R}$, any $m \leq n$, and any $H \subset [d]$ with size $d/2$. If each Q_{H,R_i} is (ε, δ)-differentially private, then there exists a value $\mu \in [-1, +1]$ such that the*

statistical distance between $\vec{R}(\mathbf{P}_{H,\mu})$ *and* $\mathrm{Manip}\Big(\vec{R}, \mathbf{U}, M^{\vec{R},H}_{m,n}\Big)$ *is at most* $1/10 + 2n\delta$ *even though*

$$\|\mathbf{U} - \mathbf{P}_{H,\mu}\|_1 = \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \frac{m}{2n} \tag{4.2}$$

Proof. By Claim 4.4, $\vec{Q}_H(\mathbf{Rad}(0)) = \vec{R}(\mathbf{P}_{H,0})$. Note that $\mathbf{P}_{H,0} = \mathbf{U}$. By Claim 4.5, $\mathrm{Manip}(\vec{R}, \mathbf{P}_{H,\mu}, M^{\vec{R},H}_{d,m,n})$ is identical to $\mathrm{Manip}\Big(\vec{Q}_H, \mathbf{Rad}(\mu), M^{\vec{Q}_H}_{m,n}\Big)$. So it will suffice to bound the statistical distance between $\vec{Q}_H(\mathbf{Rad}(0))$ and $\mathrm{Manip}\Big(\vec{Q}_H, \mathbf{Rad}(\mu), M^{\vec{Q}_H}_{m,n}\Big)$ for some choice of μ. But Lemma 3.3 implies that for $\mu = \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \frac{m}{2n}$, the distance is $1/10 + 2n\delta$.

It remains to prove (4.2). When sampling $x \sim \mathbf{P}_{H,\mu}$, the probability that $x = h$ is $\frac{1+\mu}{d}$ for each $h \in H$ and $\frac{1-\mu}{d}$ for each $h \notin H$. Hence,

$$\begin{aligned}\|\mathbf{U} - \mathbf{P}_{H,\mu}\|_1 &= \frac{d}{2} \cdot \left|\frac{1}{d} - \frac{1+\mu}{d}\right| + \frac{d}{2} \cdot \left|\frac{1}{d} - \frac{1-\mu}{d}\right| \\ &= \mu = \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \frac{m}{2n}\end{aligned}$$

This concludes the proof. □

4.2.2. *Analysis for random* H. Here, we analyze what a randomly selected H entails for the privacy parameters (ε', δ) of all Q_{H,R_i}. The analysis will essentially imply that we launch a powerful attack with constant probability

Lemma 4.7. *Fix any* $\varepsilon \in (0,1)$ *and any* $\vec{R}$ *where each* $R_i : [d] \to \mathcal{Y}$ *is* ε*-differentially private. There are constants* c_0, c_1 *such that, if* $d > c_0 \cdot (e^\varepsilon - 1)^2 \log n$ *and* H *is drawn uniformly from all subsets of* $[d]$ *with size* $d/2$*, then the following holds with probability* $> 2/3$ *over the randomness of* H*: every* Q_{H,R_i} *specified by Algorithm 2 is* $(\varepsilon', 1/360n)$*-differentially private, where*

$$\varepsilon' = \varepsilon \cdot \sqrt{\frac{c_1 \log n}{d}}$$

This statement follows from arguments made in Appendix A.2. From Lemmas 4.6 and 4.7, we see that $M^{\vec{R},H}_{d,m,n}$ is $(\approx m\sqrt{d}/\varepsilon n, 1/9)$-powerful with probability $\geq 2/3$ over H.

Lemma 4.8. *There are constants* c_0, c_1 *and a value* $\mu \in [-1, +1]$ *such that, for any* $n > m > 18$*,* $d > c_0 \cdot (e^\varepsilon - 1)^2 \log n$*, and* $\varepsilon \in (0,1)$*, the following holds: against any* ε*-locally private protocol* $\Pi = (\vec{R}, A)$*, the attack* $M^{\vec{R}}_{d,m,n}$ *chooses* H *with probability* $> 2/3$ *such that the statistical distance between* $\vec{R}(\mathbf{P}_{H,\mu})$ *and* $\mathrm{Manip}\Big(\vec{R}, \mathbf{U}, M^{\vec{R},H}_{d,m,n}\Big)$ *is at most* $1/9$ *even though*

$$\|\mathbf{U} - \mathbf{P}_{H,\mu}\|_1 \geq \frac{c_1 \cdot m\sqrt{d}}{\varepsilon n \sqrt{\log n}}$$

In the special case where there are $O(1)$ distinct randomizers that each output $O(1)$-bit messages, we can obtain an alternate version of Lemma 4.7 without the $\log n$ factor. We perform this analysis in Appendix A.3. We will focus on the present version of Lemma 4.7 to maintain full generality.

4.3. **Applications to Testing and Estimation.** From Lemma 4.8, we obtain lower bounds on how well the manipulation attack fares against protocols for uniformity testing and estimation.

Theorem 4.9. *There are constants c_0, c_1 such that, for any $n > m > 18$, $d > c_0 \cdot (e^\varepsilon - 1)^2 \log n$, $\varepsilon \in (0,1)$, the following holds for any ε-locally private uniformity testing protocol $\Pi = (\vec{R}, A)$: if $\mathbb{P}[\Pi(\mathbf{P}) = \text{“not uniform”}] \geq 95/100$ for all*

$$\|\mathbf{U} - \mathbf{P}\|_1 \geq \frac{c_1 \cdot m\sqrt{d}}{\varepsilon n \sqrt{\log n}} \tag{4.3}$$

then

$$\mathbb{P}\Big[\mathrm{Manip}\Big(\Pi, \mathbf{U}, M^{\vec{R}}_{d,m,n}\Big) = \text{“not uniform”}\Big] > 1/2$$

Theorem 4.10. *There are constants c_0, c_1 such that, for any $n > m > 18$, $d > c_0 \cdot (e^\varepsilon - 1)^2 \log n$, $\varepsilon \in (0,1)$, the following holds for any (ε, δ)-locally private for estimating distributions over $[d]$: if $\mathbb{P}[\|\Pi(\mathbf{P}) - \mathbf{P}\|_1 < \alpha] > 95/100$ for all distributions $\mathbf{P}$ and $\alpha \leq \frac{c_1 \cdot m\sqrt{d}}{\varepsilon n \sqrt{\log n}}$, then*

$$\mathbb{P}\Big[\Big\|\mathrm{Manip}\Big(\Pi, \mathbf{U}, M^{\vec{R}}_{d,m,n}\Big) - \mathbf{U}\Big\|_1 \geq \alpha\Big] > 1/2.$$

5. Protocols with Nearly Optimal Robustness to Manipulation

In this section, we consider a number of well-studied problems in local privacy and identify specific protocols from the literature with optimal robustness to manipulation (i.e. matching the lower bounds implied by our attacks). As discussed in the introduction, most of these problems can be cast as accurate mean estimation of bounded vectors. But we also study robust protocols for uniformity testing and heavy hitters (in Section 5.3 and Appendix C, respectively).

5.1. **Warmup: Mean Estimation for Binary Data.** As a warmup, we analyze the randomized response protocol in the presence of manipulation. The protocol is defined by the local randomizer $R^{\mathtt{RR}}_\varepsilon$ and aggregator $A^{\mathtt{RR}}_{n,\varepsilon}$ as follows:

$$R^{\mathtt{RR}}_\varepsilon(x) := \begin{cases} \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot x & \text{with probability } \frac{e^\varepsilon}{e^\varepsilon+1} \\ -\frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot x & \text{with probability } \frac{1}{e^\varepsilon+1} \end{cases}$$
$$A^{\mathtt{RR}}_{n,\varepsilon}(\vec{y}) := \tfrac{1}{n}\textstyle\sum_{i=1}^n y_i$$

We bound the error of this protocol by $O(\frac{1}{\varepsilon}(\frac{1}{\sqrt{n}} + \frac{m}{n})$, which matches the lower bound of Theorem 3.4 up to constants.

Theorem 5.1. *For any positive integers $m \leq n$, any $\varepsilon > 0$, any $\vec{x} \in \{0,1\}^n$, any manipulation adversary M, and any $\beta > 0$, with probability $\geq 1 - \beta$, we have*

$$\left|\mathrm{Manip}_{m,n}(\mathit{RR}_{\varepsilon,n}, \vec{x}, M) - \tfrac{1}{n}\textstyle\sum_{i=1}^n x_i\right| < \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \left(\sqrt{\tfrac{2}{n}\ln\tfrac{2}{\beta}} + \tfrac{2m}{n}\right)$$

Proof. Consider an execution of $\mathrm{Manip}(\mathtt{RR}_{\varepsilon,n}, \vec{x}, M)$. Let C be the set of corrupted users, let $y_1, \ldots, y_n$ be the messages sent in the protocol and let $\underline{\vec{y}}$ be the messages that would have been sent in an honest execution (so $\underline{y}_i = y_i$ for every $i \notin C$). Let $z = \frac{1}{n}\sum_{i=1}^n y_i$ be the output of the aggregator.

We can break up the error into two components, one corresponding to the error of the honest execution and one corresponding to the error introduced by manipulation.

$$\begin{aligned}
&\left|\tfrac{1}{n}\textstyle\sum_{i\in[n]} y_i - \tfrac{1}{n}\sum_{i\in[n]} x_i\right| \\
&= \left|\tfrac{1}{n}\textstyle\sum_{i\in[n]} y_i - \tfrac{1}{n}\sum_{i\in[n]} \underline{y}_i + \tfrac{1}{n}\sum_{i\in[n]} \underline{y}_i - \tfrac{1}{n}\sum_{i\in[n]} x_i\right| \\
&\le \left|\tfrac{1}{n}\textstyle\sum_{i\in[n]} y_i - \tfrac{1}{n}\sum_{i\in[n]} \underline{y}_i\right| + \left|\tfrac{1}{n}\textstyle\sum_{i\in[n]} \underline{y}_i - \tfrac{1}{n}\sum_{i\in[n]} x_i\right| \\
&= \underbrace{\left|\tfrac{1}{n}\textstyle\sum_{i\in C} y_i - \underline{y}_i\right|}_{\text{manipulation}} + \underbrace{\left|\tfrac{1}{n}\textstyle\sum_{i\in[n]} \underline{y}_i - \tfrac{1}{n}\sum_{i\in[n]} x_i\right|}_{\text{honest execution}}
\end{aligned}$$

Since each message in the protocol is either $\frac{e^\varepsilon+1}{e^\varepsilon-1}$ or $-\frac{e^\varepsilon+1}{e^\varepsilon-1}$, we have $|y_i - \underline{y}_i| \le 2 \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1}$. Thus, the manipulation term is bounded by $\frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \frac{2m}{n}$ with probability 1.

For the error of the honest execution, note that $\mathbb{E}[\underline{y}_i] = x_i$ and $\frac{1}{n}\sum_{i\in[n]} \underline{y}_i$ is an average of n independent random variables bounded to a range of width $2 \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1}$. Thus, by Hoeffding's inequality, the second term is bounded by $\frac{e^\varepsilon+1}{e^\varepsilon-1}\sqrt{\frac{2\ln(2/\beta)}{n}}$ with probability at least $1-\beta$. □

Our analysis of richer protocols has the same structure. We construct the protocol so that each message y_i gives an unbiased estimate of x_i, and the aggregation computes the mean of the messages. We then isolate the effect of the manipulation from that of an honest execution. Finally, we bound the influence of m messages on the output of the protocol. For richer protocols the analysis of the final step will become more involved.

5.2. **Mean Estimation.** We consider vector-valued data in $\mathbb{R}^d$. For any $p \ge 1$, $\|x\|_p := (\sum_{j=1}^d |x_j|^p)^{1/p}$ denotes the standard ℓ_p norm and B_p^d denotes the ℓ_p unit ball in $\mathbb{R}^d$. As is standard $\|x\|_\infty = \max_{j\in[d]} |x_j|$ is the ℓ_∞ norm and B_∞^d is the ℓ_∞ unit ball. In this section, we study instances of the general ℓ_p/ℓ_q *mean estimation problem*: given data $x_1, \ldots, x_n \in B_p^d$, output some $\hat{\mu}$ such that $\left\|\hat{\mu} - \frac{1}{n}\sum_i x_i\right\|_q$ is as small as possible.

5.2.1. ℓ_∞/ℓ_∞ *estimation (Counting Queries).* In this problem, each user has data $x_i \in B_\infty^d$ and the goal is to obtain a vector $\hat{\mu}$ such that $\left\|\hat{\mu} - \frac{1}{n}\sum x_i\right\|_\infty$ is as small as possible. We consider the following protocol $\mathtt{EST}\infty = (R^{\mathtt{EST}\infty}, n, A^{\mathtt{EST}\infty})$, which is known to have optimal error absent manipulation.

(1) Using public randomness, we partition users into d groups each of size n/d. Intuitively, we are assigning each group to one coordinate.
(2) For each group j, each user i in group j reports the message $y_i \leftarrow R^{\mathtt{RR}}(x_{i,j})$
(3) For each group j, the aggregator computes the average of the messages from group j to obtain $\hat{\mu}_j \approx \frac{1}{n}\sum_i x_{i,j}$. The aggregator reports $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_d)$

If the adversary's corruptions are oblivious to the public partition, then we show that there are $\approx m/d$ corrupt users in each group of size n/d. By our analysis of randomized response, the adversary can introduce at most $\approx \frac{m/d}{\varepsilon n/d} = \frac{m}{\varepsilon n}$ error in any single coordinate.

Theorem 5.2. *For any $\varepsilon \in (0,1)$, any positive integers $m \le n$, any $x_1, \dots, x_n \in B_\infty^d$, and any* public-string-oblivious *adversary M, with probability $\ge 99/100$, we have*

$$\left\|\mathrm{Manip}_{m,n}(\mathit{EST}\infty_\varepsilon, \vec{x}, M) - \tfrac{1}{n}\textstyle\sum_{i=1}^n x_i\right\|_\infty = O\left(\sqrt{\tfrac{d\log d}{\varepsilon^2 n}} + \tfrac{m}{\varepsilon n}\right)$$

Observe that the dependence on m matches that of the lower bound in Theorem 3.4 for Bernoulli estimation. We give the complete details of the protocol in Appendix B.1.

5.2.2. *ℓ_1/ℓ_∞ Estimation (Histograms).* In this problem, each user i has data $x_i \in B_1^d$ and the objective is a $\hat{\mu}$ such that $\left\|\hat{\mu} - \frac{1}{n}\sum_{i=1}^n x_i\right\|_\infty$ is as small as possible. To simplify the discussion, we focus on the special case where user i has data $x_i \in [d]$; here, the output is a histogram. Define $\mathit{freq}(j, \vec{x}) := \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{x_i = j\}}$ and $\mathit{freq}(\vec{x}) := (\mathit{freq}(1,\vec{x}), \dots, \mathit{freq}(1,\vec{x}))$. The objective is a vector $\hat{\mu}$ such that $\|\hat{\mu} - \mathit{freq}(\vec{x})\|_\infty$ is as small as possible.

We consider the following protocol $\mathtt{HST}_\varepsilon$, [3] which is known to have optimal error absent manipulation:

(1) For each user i, independently sample a uniform public vector $\vec{s}_i \in \{\pm 1\}^d$.
(2) Each user i reports the message $y_i \leftarrow R_\varepsilon^{\mathtt{RR}}(s_{i,x_i})$.
(3) The aggregator receives messages $y_1, \dots, y_n$ and outputs $\hat{\mu} \leftarrow \frac{1}{n}\sum_{i=1}^n y_i \cdot \vec{s}_i$.

Theorem 5.3. *For any $\varepsilon \in (0,1)$, any positive integers $m \le n$, any $x_1, \dots, x_n \in [d]$, and any adversary M, with probability at least $99/100$, we have*

$$\left\|\mathrm{Manip}_{m,n}(\mathit{HST}_\varepsilon, \vec{x}, M) - \mathit{freq}(\vec{x})\right\|_\infty = O\left(\sqrt{\frac{\log d}{\varepsilon^2 n}} + \frac{m}{\varepsilon n}\right)$$

Proof Sketch. Identically to the proof of Theorem 5.1, we partition the error contributed by the honest and corrupt users. Let $\vec{y}$ be the messages sent in the protocol and let $\underline{\vec{y}}$ be the messages that would have been sent in an honest execution. Below, $\sum_i \dots$ will be short for $\sum_{i=1}^n \dots$. We can write

$$\begin{aligned}&\left\|\tfrac{1}{n}\textstyle\sum_i \vec{s}_i - \mathit{freq}(\vec{x})\right\|_\infty \\ &= \underbrace{\max_{j\in[d]}\left[\tfrac{1}{n}\textstyle\sum_{i\in C}(y_i - \underline{y}_i)s_{i,j}\right]}_{\text{manipulation}} + \underbrace{\max_{j\in[d]}\left|\tfrac{1}{n}\textstyle\sum_i \underline{y}_i s_{i,j} - \mathbb{1}_{\{x_i=j\}}\right|}_{\text{honest execution}}\end{aligned}$$

To bound the error from the manipulation, note that messages have magnitude $\frac{e^\varepsilon+1}{e^\varepsilon-1} = \Theta(1/\varepsilon)$. Hence, the bias at any coordinate j is at most $O(m/\varepsilon n)$ with probability 1.

We now bound the error of the honest execution. If $x_i = j$, the expectation of $\underline{y}_i s_{i,j}$ is 1. Otherwise, the expectation is 0 because of pairwise independence. Hence, the honest execution has 0 expected error. Because messages have magnitude $\Theta(1/\varepsilon)$, Hoeffding's inequality and a union bound imply that no frequency estimate is more than $O(\sqrt{\log d/\varepsilon^2 n})$ from $\mathit{freq}(j, \vec{x})$ with probability $\ge 99/100$. □

[3] In Bassily et al. (2017) the protocol is called `ExplicitHist`.

A slightly more general protocol can be used to obtain the same result for ℓ_1/ℓ_∞ estimation.

Theorem 5.4. *For any $\varepsilon \in (0,1)$, there is an ε-locally private protocol $\texttt{EST1}_\varepsilon$ such that for any positive integer n, any $x_1, \dots, x_n \in B_1^d$, and any adversary M, with probability $\geq 99/100$, we have*

$$\left\|\mathrm{Manip}_{m,n}(\texttt{EST1}_\varepsilon, \vec{x}, M) - \tfrac{1}{n}\textstyle\sum_{i=1}^n x_i\right\|_\infty = O\left(\sqrt{\frac{\log d}{\varepsilon^2 n}} + \frac{m}{\varepsilon n}\right)$$

We give the complete details of $\texttt{EST1}_\varepsilon$ in Appendix B.2. Observe that its manipulation error matches that of Bernoulli estimation (Theorem 3.4).

5.2.3. *ℓ_1/ℓ_1 Estimation (Frequency Estimation).* In this problem, each user i has data $x_i \in B_1^d$ and the objective is a $\hat{\mu}$ such that $\left\|\hat{\mu} - \frac{1}{n}\sum_{i=1}^n x_i\right\|_1$ is as small as possible. Because this problem and the ℓ_1/ℓ_∞ problem have the same data type, we consider the same protocols but change the analysis to upper bound ℓ_1 error.

Theorem 5.5. *For any $\varepsilon \in (0,1)$, any positive integer n, any $x_1, \dots, x_n \in [d]$, and any adversary M, with probability at least $99/100$, we have*

$$\left\|\mathrm{Manip}_{m,n}(\texttt{HST}_\varepsilon, \vec{x}, M) - \tfrac{1}{n}\textstyle\sum_{i=1}^n x_i\right\|_1 = O\left(\sqrt{\frac{d^2 \log n}{\varepsilon^2 n}} + \frac{m\sqrt{d \log n}}{\varepsilon n}\right)$$

Proof Sketch. Identically to the proof of Theorem 5.1, we partition the error contributed by the honest and corrupt users. Let $\vec{y}$ be the messages sent in the protocol and let $\underline{\vec{y}}$ be the messages that would have been sent in an honest execution. Let $S \in \{\pm 1\}^{d \times n}$ be the matrix whose columns are $\vec{s}_1, \dots, \vec{s}_n$, and $S_C \in \{\pm 1\}^{d \times |C|}$ be the submatrix consisting only of columns corresponding to users $i \in C$. Then we can write

$$\begin{aligned} &\left\|\tfrac{1}{n}\textstyle\sum_{i=1}^n y_i \vec{s}_i - \mathit{freq}(\vec{x})\right\|_1 \\ &= \underbrace{\left\|\tfrac{1}{n} S_C(\vec{y}_C - \underline{\vec{y}}_C)\right\|_1}_{\text{manipulation}} + \underbrace{\sum_{j \in [d]} \left|\tfrac{1}{n}\textstyle\sum_{i=1}^n \underline{y}_i s_{i,j} - \mathbb{1}_{\{x_i = j\}}\right|}_{\text{honest execution}} \end{aligned}$$

To bound the error from the honest execution, observe that the expectation and variance are $O(\sqrt{1/\varepsilon^2 n})$ and $O(1/\varepsilon n)$, respectively, for any term in the outer sum. Hence, error has magnitude $O(\sqrt{d^2/\varepsilon^2 n})$ with probability $\geq 199/200$.

To bound the error from the manipulation, we will use bounds on the singular values of the random matrix S_C. As a shorthand, let $c_\varepsilon = \frac{e^\varepsilon+1}{e^\varepsilon-1}$. Then we have

$$\begin{aligned}
\left\|\tfrac{1}{n}S_C(\vec{y}_C - \underline{\vec{y}}_C)\right\|_1 &\le \tfrac{1}{n}\max_{C\subseteq[n]}\left\|S_C(\vec{y}_C - \underline{\vec{y}}_C)\right\|_1 \\
&\le \frac{2}{n}\max_{\substack{C\subseteq[n]\\|C|=m}}\ \max_{\vec{y}_C\in\{-c_\varepsilon,c_\varepsilon\}^m}\|S_C\vec{y}_C\|_1 \\
&= \frac{2}{n}\max_{\substack{C\subseteq[n]\\|C|=m}}\ \max_{\substack{\vec{y}_C\in\mathbb{R}^m\\\|\vec{y}\|_2\le c_\varepsilon\sqrt{m}}}\|S_C\vec{y}_C\|_1 \\
&\le \frac{c_\varepsilon\sqrt{md}}{n}\max_{\substack{C\subseteq[n]\\|C|=m}}\ \max_{\substack{\vec{y}_C\in\mathbb{R}^m\\\|\vec{y}\|_2\le 1}}\|S_C\vec{y}_C\|_2 \\
&= \frac{c_\varepsilon\sqrt{md}}{n}\max_{\substack{C\subseteq[n]\\|C|=m}}\|S_C\|_2
\end{aligned}$$

where $\|S_C\|_2$ denotes the largest singular value (operator norm) of S_C. Since each matrix $S_C \in \{\pm1\}^{d\times m}$ is uniformly random, we can use bounds on the singular values of random matrices.

Lemma 5.6 (see e.g. the textbook by Tao (2012))**.** *For any $k \in \mathbb{R}_+$ larger than an absolute constant and a matrix $S_C \in \mathbb{R}^{d\times m}$ whose entries are sampled independently and identically, the following holds with probability $\ge 1-\exp(-k(d+m))$ over the randomness of S_C.*

$$\|S_C\|_2 = O(\sqrt{kd}+\sqrt{km})$$

The adversary has $\binom{n}{m} \le \exp(m\ln n)$ choices of corruptions C. By a union bound over that set, we have with probability at least $1-\exp(m\ln n - k(m+d))$

$$\begin{aligned}
\left\|\tfrac{1}{n}S_C(\vec{y}_C - \underline{\vec{y}}_C)\right\|_1 &\le \frac{c_\varepsilon\sqrt{md}}{n}\cdot O(\sqrt{kd}+\sqrt{km}) \\
&= O\left(\sqrt{\frac{d^2k}{\varepsilon^2 n}} + \frac{m\sqrt{dk}}{\varepsilon n}\right)
\end{aligned}$$

For $k = O(\log n)$, the bound holds with probability at least $199/200$. □

A slightly more general protocol can be used to obtain the same result for ℓ_1/ℓ_1 estimation.

Theorem 5.7. *For any $\varepsilon \in (0,1)$, there is an ε-locally private protocol* `EST1` *such that for any positive integer n, any $x_1,\dots,x_n \in (B_1^d)^n$, and any adversary M, with probability $\ge 99/100$, we have*

$$\left\|\mathrm{Manip}_{m,n}(\texttt{EST1}_\varepsilon, \vec{x}, M) - \tfrac{1}{n}\textstyle\sum_{i=1}^n x_i\right\|_1 = O\left(\sqrt{\frac{d^2\log n}{\varepsilon^2 n}} + \frac{m\sqrt{d\log n}}{\varepsilon n}\right)$$

Observe that the manipulation error matches the lower bound in Theorem 4.10, up to a logarithmic factor.

5.2.4. ℓ_2/ℓ_2 *Estimation.* In this problem, each user i has data $x_i \in B_2^d$ and the objective is a $\hat{\mu}$ such that $\left\|\hat{\mu} - \frac{1}{n}\sum_{i=1}^n x_i\right\|_2$ is as small as possible.

Consider the following protocol `EST2` adapted from (Duchi et al., 2016, Section 4.2.3):

(1) For each user i, we sample $\vec{s}_i \in \mathbb{R}^d$ uniformly at random from the surface of B_2^d.
(2) Each user i computes $w_i \leftarrow \operatorname{sgn}(\vec{s}_i \cdot x_i)$ and then reports $y_i \leftarrow R_\varepsilon^{\mathtt{RR}}(w_i)$ to the aggregator
(3) The aggregator receives the messages $y_1, \ldots, y_n$ and outputs $\vec{z} \leftarrow \frac{c\sqrt{d}}{n}\sum_{i=1}^n y_i \vec{s}_i$ for some constant $c > 0$.

Theorem 5.8. *For any $\varepsilon \in (0,1)$, any positive integer n, any $x_1, \ldots, x_n \in B_2^d$, and any adversary M, with probability $\geq 99/100$, we have*

$$\left\|\mathrm{Manip}_{m,n}(\textit{EST2}_\varepsilon, \vec{x}, M) - \tfrac{1}{n}\textstyle\sum_{i=1}^n x_i\right\|_2 = O\left(\sqrt{\tfrac{d\log n}{\varepsilon^2 n}} + \tfrac{m\sqrt{\log n}}{\varepsilon n}\right)$$

Proof Sketch. Identically to the proof of Theorem 5.1, we partition the error contributed by the honest and corrupt users. Let $S \in \{\pm 1\}^{d\times n}$ be the matrix whose columns are $\vec{s}_1, \ldots, \vec{s}_n$, and $S_C \in \{\pm 1\}^{d\times |C|}$ be the submatrix consisting only columns corresponding to users $i \in C$. Below, $\sum_i$ stands for $\sum_{i=1}^n$ Then we can write

$$\begin{aligned}
&\left\|\tfrac{c\sqrt{d}}{n}\textstyle\sum_i y_i\vec{s}_i - \tfrac{1}{n}\sum_i x_i\right\|_2 \\
&= \underbrace{\left\|\tfrac{c\sqrt{d}}{n} S_C(\vec{y}_C - \underline{\vec{y}}_C)\right\|_2}_{\text{manipulation}} + \underbrace{\left\|\tfrac{c\sqrt{d}}{n}\textstyle\sum_i \underline{y}_i\vec{s}_i - \tfrac{1}{n}\sum_{i=1}^n x_i\right\|_2}_{\text{honest execution}}
\end{aligned}$$

Corollary 4 in Duchi et al. (2016) implies that the error introduced by the honest execution of the protocol is $O(\sqrt{d/\varepsilon^2 n})$ with probability $\geq 299/300$.

To bound the error from the manipulation, we will again use bounds on the singular values of the random matrix S_C. As a shorthand, let $c_\varepsilon = \frac{e^\varepsilon+1}{e^\varepsilon-1}$. Then we have

$$\begin{aligned}
&\left\|\tfrac{c\sqrt{d}}{n} S_C(\vec{y}_C - \underline{\vec{y}}_C)\right\|_2 \\
&\leq \tfrac{c\sqrt{d}}{n}\max_{C\subseteq[n]}\left\|S_C(\vec{y}_C - \underline{\vec{y}}_C)\right\|_2 \\
&\leq \tfrac{2c\sqrt{d}}{n}\max_{\substack{C\subseteq[n]\\|C|=m}}\ \max_{\vec{y}_C\in\{-c_\varepsilon,c_\varepsilon\}^m}\|S_C\vec{y}_C\|_2 \\
&= \tfrac{2c\sqrt{d}}{n}\max_{\substack{C\subseteq[n]\\|C|=m}}\ \max_{\substack{\vec{y}_C\in\mathbb{R}^m\\\|\vec{y}\|_2\leq c_\varepsilon\sqrt{m}}}\|S_C\vec{y}_C\|_2 \\
&\leq \tfrac{2cc_\varepsilon\sqrt{md}}{n}\max_{\substack{C\subseteq[n]\\|C|=m}}\ \max_{\substack{\vec{y}_C\in\mathbb{R}^m\\\|\vec{y}\|_2\leq 1}}\|S_C\vec{y}_C\|_2
\end{aligned} \tag{5.1}$$

For any $i \in C$, consider the random variable $\vec{s}_i\,' \sim N(0, I_{d\times d})$. The column vector $\vec{s}_i$ is identically distributed with $\frac{\vec{s}_i\,'}{\|\vec{s}_i\,'\|_2}$. By standard concentration arguments, there is a constant c' such that $\min_i\|\vec{s}_i\,'\|_2^2 \geq d - c'\sqrt{d\ln m}$ with probability $\geq 299/300$. In the case where $d < 4(c')^2\ln m$, we bound the error by $2cc_\varepsilon m\sqrt{d}/n = O(m\sqrt{\log n}/\varepsilon n)$. Otherwise, when

$d > 4(c')^2 \ln m$, we have $\min_i \|\vec{s}_i\,'\|_2^2 > d/2$. Hence,

$$\begin{aligned}(5.1) &\le \frac{2cc_\varepsilon\sqrt{md}}{n} \max_{\substack{C\subseteq[n]\\|C|=m}} \max_{\substack{\vec{y}_C\in\mathbb{R}^m\\\|\vec{y}\|_2\le 1}} \max_{i\in C} \frac{1}{\|\vec{s}_i\,'\|_2} \left\|S'_C\vec{y}_C\right\|_2 \\ &\le \frac{cc_\varepsilon\sqrt{8m}}{n} \max_{\substack{C\subseteq[n]\\|C|=m}} \max_{\substack{\vec{y}_C\in\mathbb{R}^m\\\|\vec{y}\|_2\le 1}} \left\|S'_C\vec{y}_C\right\|_2 \\ &= \frac{cc_\varepsilon\sqrt{8m}}{n} \max_{\substack{C\subseteq[n]\\|C|=m}} \left\|S'_C\right\|_2\end{aligned}$$

We apply Lemma 5.6 then choose $k = O(\ln n)$ to bound $\|S'_C\|_2$ by $O(\sqrt{d\ln n} + \sqrt{m\ln n})$ with probability $\ge 299/300$. A union bound completes the proof. □

Observe that the manipulation error matches that of Bernoulli estimation (Theorem 3.4) up to a logarithmic factor.

5.3. **Uniformity Testing.** In this problem, each user has data $x_i \in [d]$ sampled from a distribution $\mathbf{P}$. If $\mathbf{P} = \mathbf{U}$, then a protocol for this problem should output "uniform" with probability $\ge 99/100$. If $\|\mathbf{P} - \mathbf{U}\|_1 > \alpha$, then it should output "not uniform" with probability $\ge 99/100$. Smaller values of α are desirable.

We consider the RAPTOR protocol, introduced by Acharya et al. (2019). It divides users into G groups each of size n/G (where G is a parameter). In each group g,

(1) Sample public set $S \in \{S \subset [d] \mid |S| = d/2\}$ uniformly at random.
(2) Each user assigns $x'_i \leftarrow +1$ if $x_i \in S$ and otherwise $x'_i \leftarrow -1$
(3) Each user i reports $y_i \leftarrow R^{\mathsf{RR}}_\varepsilon(x'_i)$ to the aggregator
(4) The aggregator computes the average of the messages: $\hat{\mu}_g \leftarrow \frac{G}{n}\sum y_i$.

If there is some $\hat{\mu}_g \gtrapprox \sqrt{\frac{1}{\varepsilon^2 n}} + \frac{m}{\varepsilon n}$, the aggregator reports "not uniform." Otherwise, it reports "uniform."

Theorem 5.9. *There is a choice of parameter G such that, for any $\varepsilon \in (0,1)$, any positive integers $m \le n$, and any adversary M, the following holds with probability $\ge 99/100$*

$$\mathrm{Manip}_{m,n}(\textit{RAPTOR}_\varepsilon, \mathbf{U}, M) = \text{"uniform"}$$

and, when $\|\mathbf{P} - \mathbf{U}\|_1 \ge \alpha$ for some $\alpha = O\left(\sqrt{\frac{d}{\varepsilon^2 n}} + \frac{m\sqrt{d}}{\varepsilon n}\right)$, the following also holds with probability $\ge 99/100$

$$\mathrm{Manip}_{m,n}(\textit{RAPTOR}_\varepsilon, \mathbf{P}, M) = \text{"not uniform"}$$

Proof Sketch. Consider any $g \in [G]$. When $\|\mathbf{P} - \mathbf{U}\|_1 \ge \sqrt{10d}\cdot\alpha$, a lemma by Acharya et al. (2019) implies that, with at least some constant probability over the randomness of S, $\left|\mathbb{P}_{x\sim\mathbf{P}}[x \in S] - \frac{1}{2}\right| \gtrapprox \alpha$. For $\alpha \gtrapprox \sqrt{G/\varepsilon^2 n} + mG/\varepsilon n$, $\mathtt{RR}_\varepsilon$ will provide an estimate of $\mathbb{P}_{x\sim\mathbf{P}}[x \in S]$ that is larger than $\frac{1}{2} + \alpha/2$. But when $\mathbf{P} = \mathbf{U}$, the protocol will give an estimate of $\mathbb{P}_{x\sim\mathbf{P}}[x \in S]$ that is less than $\frac{1}{2} + \alpha/2$. This means there is a threshold test that has a constant probability of succeeding. The G repetitions serve to increase the success probability to 99/100. This completes the proof. □

Observe that the bound $\alpha = O(\frac{m\sqrt{d}}{\varepsilon n} + \sqrt{\frac{d}{\varepsilon^2 n}})$ matches the lower bound of Theorem 4.9 up to logarithmic factors. We give the complete details in Appendix B.3.

6. Accurate Protocols that are Not Robust

In this section, we demonstrate that there exist protocols with optimal error absent manipulation ($m = 0$) that perform quite poorly in the presence of manipulation ($m > 0$). Thus, a careful choice of protocols is necessary to achieve optimal robustness.

Intuitively, the protocols in Section 5 achieve optimal robustness because they use public randomness to significantly constrain the choices of the corrupted users. Allowing users to generate the randomness themselves has no effect on the protocol absent manipulation but we argue that the protocol becomes much less robust.

We can sketch an example of this phenomenon for frequency estimation, although essentially the same phenomenon arises in all of the problems we study. Consider the following variant of the frequency estimation protocol:

(1) Each user chooses a uniformly random vector $\vec{s}_i \in \{\pm 1\}^d$.
(2) Each user samples $\gamma_i \leftarrow R_\varepsilon^{\mathtt{RR}}(\vec{s}_{i,x_i})$ and reports the message $\vec{y}_i \leftarrow \gamma_i \vec{s}_i \in \{\pm\frac{e^\varepsilon+1}{e^\varepsilon-1}\}^d$.
(3) The aggregator outputs $\hat{\mu} \leftarrow \frac{1}{n}\sum_{i=1}^n \vec{y}_i$.

One can verify that when all users follow the protocol honestly, the distribution of the output $\hat{\mu}$ is identical to that of the protocol $\mathtt{HST}_\varepsilon$. Therefore, when users are honest, with high probability we have $\|\hat{\mu} - \mathit{freq}(\vec{x})\|_1 = O(\sqrt{d^2/\varepsilon^2 n})$.

However, because the adversary can have the corrupted users report arbitrary vectors in $\{\pm\frac{e^\varepsilon+1}{e^\varepsilon-1}\}^d$, an adversary who corrupts the first m users can introduce error on the order of

$$\max_{\vec{y}_1,\ldots,\vec{y}_m \in \{\pm 1\}^d} \left\| \frac{1}{n}\sum_{i=1}^m \left(\tfrac{e^\varepsilon+1}{e^\varepsilon-1}\right)\vec{y}_i \right\|_1 = \left(\tfrac{e^\varepsilon+1}{e^\varepsilon-1}\right)\cdot\tfrac{md}{n} = \Omega\big(\tfrac{md}{\varepsilon n}\big)$$

In contrast, when we use the protocol $\mathtt{HST}_\varepsilon$, we were able to show that the adversary could only introduce error $O(\frac{m\sqrt{d}}{\varepsilon n})$.

7. Experiments

In this section we give a basic set of experiments with our attack against the natural frequency estimation protocol HST, which we showed to be optimally robust to manipulation (Theorem 5.5). These experiments validate our theoretical analysis by showing that—at least for the protocol HST—the vulnerability to manipulation depends significantly on the dimension of the input domain. The experiments also indicate that the concrete error introduced by the attack against the natural protocol HST is significantly larger than what our worst-case analysis guarantees against arbitrary protocols.

In our experiments, we generate data from the uniform distribution over the domain $\{1, \ldots, d\}$ and measure the ℓ_1 error of the protocol HST. In our experiments, we fix $n = 2\times 10^5$ and $\varepsilon = 1.0$, and vary the dimension d and the fraction of corrupted users m/n. In Figure 5, we plot the median ℓ_1 error as well as the upper and lower quartiles of the error. Table 2 gives the approximate breakdown point for varying choices of d. For purposes of concreteness, we

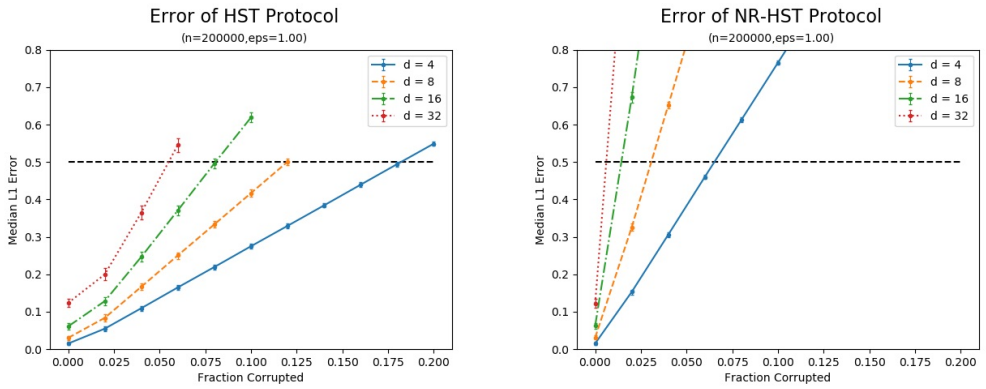

Figure 5: ℓ_1-error of the HST and NR-HST protocols for $n = 2 \times 10^5$ users, $\varepsilon = 1.0$, and various choices of dimension d and the fraction of corrupted users m/n. Each point represents the median error across 896 trials. The bars depict the 25% and 75% quantiles. The horizontal line is the breakdown point (error 0.5).

define the breakdown point as the fraction of corrupted users at which the error becomes at least 0.5, although we note that even much smaller error is likely unacceptable in applications.

We also do the same set of experiments with an alternative protocol NR-HST (for *non-robust* HST). This protocol differs from HST only in that each user samples a uniform vector $\vec{s}_i \in \{\pm 1\}^d$ themselves, and then sends $y_i \cdot \vec{s}_i$. For comparison, in HST, the user receives the vector $\vec{s}_i$ as public randomness, and only sends the single bit y_i. Note that if all users play honestly, then the distribution of the aggregator's output is *identical* to HST. However, since the corrupted users can now change how they choose $\vec{s}_i$ in addition to how they choose y_i, the protocol is much less robust to manipulation, and our experiments in Table 2 show that the protocol is much less robust to our attack.

Our final round of experiments reveal that, under a different measure of error, NR-HST is vulnerable to just a *small number* of corrupt users. The ℓ_1 norm scales with the quantity $\max_{S \subset [d]} |\sum_{j \in S} z_j - \mathit{freq}(j, \vec{x})|$, the *maximum* total error of any subset. But a data analyst may have little interest in the maximum and instead have a *target* subset, like frequencies of specific words. In Figure 6, we depict the total error of NR-HST on $S = \{1, \ldots, d/2\}$ for $n = 5 \cdot 10^4$ users. When $d = 32$, this error is under 0.05 when there are no corrupted users but it increases by around a factor of 3 when there are *only 250 corrupted users*.

8. Conclusion

This paper systematically studies *manipulation attacks* on locally differentially private protocols, in which malicious clients inject improperly generated messages into the protocol in order to influence its output. We show that vulnerability to such attacks is inherent to the model—-every noninteractive local protocol admits such attacks, and the attacks' effectiveness increases as the privacy guarantee gets stronger and, for some tasks, as the dimension of the data grows.

Our work leaves open a number of technical questions. Can interactive local protocols resist manipulation more effectively than non-interactive protocols? Can we close the

Dimension (d)	Breakdown Point (Error = 0.5)	
	`HST`	`NR-HST`
4	$\approx 18\%$	$\approx 7\%$
8	$\approx 12\%$	$\approx 3\%$
16	$\approx 8\%$	$< 2\%$
32	$\approx 5\%$	$\ll 1\%$

Table 2: Upper bounds on the breakdown point (error 0.5) of the `HST` protocol for $n = 2\times10^5$, $\varepsilon = 1.0$, and various choices of dimension d.

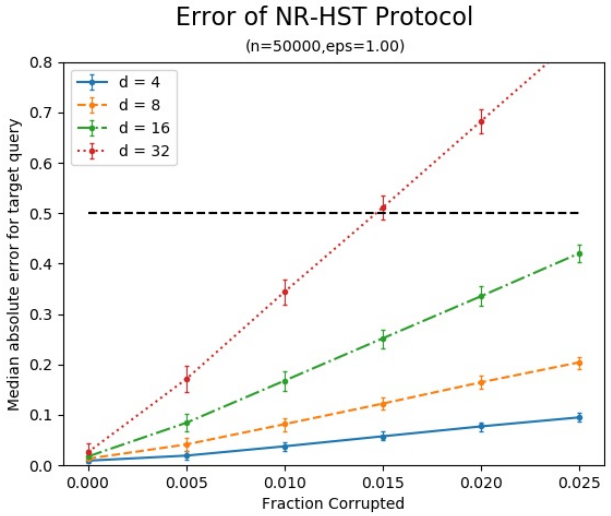

Figure 6: Error of the `NR-HST` protocol for $n = 5 \cdot 10^4$ users, $\varepsilon = 1.0$, and various choices of d and m/n. Here, error is computed by taking the sum of frequency estimation errors of $\{1, \dots d/2\}$.

few remaining gaps between upper and lower bounds in Table 1? More fundamentally, it highlights the importance of systems that collect and analyze sensitive information at scale with minimal trust requirements and strong privacy guarantees. Multiparty computation (as in Ambainis et al. (2004) and Dwork et al. (2006a)) and work on the shuffled model (Bittau et al., 2017; Cheu et al., 2019; Erlingsson et al., 2019)) are possible solutions, and other effective alternatives surely remain to be found.

Acknowledgments

AC and JU were supported by NSF grants CCF-1718088, CCF-1750640, and CNS-1816028. JU was also supported by a Google Faculty Research Award. AS was supported by NSF award CCF-1763786 and a Sloan Foundation Research Award.

Part of this work was done while the authors were visiting the Simons Institute for Theory of Computing. The authors are grateful to Henry Corrigan-Gibbs, Úlfar Erlingsson,

Vitaly Feldman, Gautam Kamath, Seth Neel, and Aaron Roth for helpful discussions, as well as to anonymous reviewers for constructive critical feedback. The authors thank Jack Doerner for help preparing the figures.

References

J. Acharya, C. L. Canonne, C. Freitag, and H. Tyagi. Test without trust: Optimal locally private distribution testing. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, AISTATS '19, pages 2067–2076. JMLR, Inc., 2019. URL http://proceedings.mlr.press/v89/acharya19b.html.

A. Ambainis, M. Jakobsson, and H. Lipmaa. Cryptographic randomized response techniques. In *Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004*, pages 425–438, 2004. doi:10.1007/978-3-540-24632-9_31.

Apple Differential Privacy Team. Learning with privacy at scale, December 2017. URL https://machinelearning.apple.com/research/learning-with-privacy-at-scale.

R. Bassily and A. D. Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 127–135, 2015. doi:10.1145/2746539.2746632.

R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science*, FOCS '14, pages 464–473, Philadelphia, PA, 2014. IEEE. doi:10.1109/FOCS.2014.56.

R. Bassily, K. Nissim, U. Stemmer, and A. G. Thakurta. Practical locally private heavy hitters. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2285–2293, 2017. URL http://papers.nips.cc/paper/6823-practical-locally-private-heavy-hitters.

A. Beimel, K. Nissim, and E. Omri. Distributed private data analysis: On simultaneously solving how and what. *CoRR*, abs/1103.2626, 2011. URL http://arxiv.org/abs/1103.2626.

A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, pages 441–459. ACM, 2017. doi:10.1145/3132747.3132769.

A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In *Proceedings of the 24th ACM Symposium on Principles of Database Systems*, PODS '05, pages 128–138. ACM, 2005. doi:10.1145/1065167.1065184.

K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1175–1191. ACM, 2017. doi:10.1145/3133956.3133982.

M. Bun, J. Nelson, and U. Stemmer. Heavy hitters and the structure of local privacy. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles*

of Database Systems, Houston, TX, USA, June 10-15, 2018, pages 435–447, 2018. doi:10.1145/3196959.3196981.

X. Cao, J. Jia, and N. Z. Gong. Data poisoning attacks to local differential privacy protocols. *arXiv preprint arXiv:1911.02046*, 2019. URL http://arxiv.org/abs/1911.02046.

A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. Distributed differential privacy via shuffling. In *Proceedings of the 38th Annual Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '19, 2019. doi:10.1007/978-3-030-17653-2_13.

J. Duchi, M. Jordan, and M. Wainwright. Local privacy and statistical minimax rates. In *IEEE 57th Annual Symposium on Foundations of Computer Science*, FOCS '13, pages 429–438, 2013a. doi:10.1109/FOCS.2013.53.

J. Duchi, M. Jordan, and M. Wainwright. Local privacy and minimax bounds: Sharp rates for probability estimation. In *Advances in Neural and Information Processing Systems 27*, NIPS '13, 2013b. URL http://papers.nips.cc/paper/5013-local-privacy-and-minimax-bounds-sharp-rates-for-probability-estimation.

J. Duchi, M. Jordan, and M. Wainwright. Minimax optimal procedures for locally private estimation. *CoRR*, abs/1604.02390, 2016. URL http://arxiv.org/abs/1604.02390.

C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, pages 486–503, 2006a. doi:10.1007/11761679_29.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006b. Springer. doi:10.1007/11681878_14.

Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the ACM Conference on Computer Security*, CCS'14, pages 1054–1067. ACM, 2014. doi:10.1145/2660267.2660348.

Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In T. M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2468–2479. SIAM, 2019. doi:10.1137/1.9781611975482.151.

A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, pages 211–222, New York, NY, USA, 2003. ACM. doi:10.1145/773153.773174.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 01621459. doi:10.2307/2282952.

J. Hsu, S. Khanna, and A. Roth. Distributed private heavy hitters. In *International Colloquium on Automata, Languages, and Programming*, pages 461–472. Springer, 2012. doi:10.1007/978-3-642-31594-7_39.

P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1376–1385, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/kairouz15.html.

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

S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *FOCS*, pages 531–540. IEEE, Oct 25–28 2008. doi:10.1109/FOCS.2008.27.

M. J. Kearns. Efficient noise-tolerant learning from statistical queries. In *STOC*, pages 392–401. ACM, May 16-18 1993. doi:10.1145/167088.167200.

T. Moran and M. Naor. Polling with physical envelopes: A rigorous analysis of a human-centric protocol. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, pages 88–108, 2006. doi:10.1007/11761679_7.

T. Tao. *Topics in Random Matrix Theory.* American Mathematical Society, 2012.

S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965. doi:10.2307/2283137.

Appendix A. Proofs for Section 4

A.1. Proof of Claim 4.2.

We restate Claim 4.2 below:

Claim A.1 (Restatement of Claim 4.2)**.** Suppose manipulation attack M is (α, γ)-powerful against $(\varepsilon, 0)$-locally private protocols. Then there is an M_δ that is $(\alpha_\delta, \gamma_\delta)$-powerful against (ε, δ)-locally private protocols, where $\alpha_\delta(\varepsilon) := \alpha(2\varepsilon)$ and $\gamma_\delta := \gamma + 2n\delta$.

Proof. We define M_δ to be the attack that, when playing against locally private protocol $\Pi = (\vec{R}, A)$, constructs $\vec{R}'$ as ensured by Lemma A.2 below and then executes attack M as if it were playing against $\Pi' = (\vec{R}', A)$. Because M is (α, γ)-powerful against ε-locally private protocols and Π' is 2ε-locally private, there is some $\mathbf{P}$ where $\|\mathbf{P} - \mathbf{U}\|_1 \geq \alpha(2\varepsilon)$ but $d_{SD}(\Pi'(\mathbf{P}), \mathrm{Manip}_{m,n}(\Pi', \mathbf{U}, M) \leq \gamma$.

We will rely on the following technical lemma:

Lemma A.2. *Every local randomizer $R : [d] \to \mathcal{Y}$ that is (ε, δ)-differentially private implies another randomizer $R' : [d] \to \mathcal{Y}$ that is 2ε-differentially private such that $d_{SD}(R(x), R'(x)) \leq \delta$ for any $x \in [d]$.*

This lemma helps us upper bound $d_{SD}(\Pi(\mathbf{P}), \mathrm{Manip}_{m,n}(\Pi, \mathbf{U}, M_\delta))$:

$$\begin{aligned}
&d_{SD}(\Pi(\mathbf{P}), \mathrm{Manip}_{m,n}(\Pi, \mathbf{U}, M_\delta)) \\
&\le d_{SD}(\Pi'(\mathbf{P}), \mathrm{Manip}_{m,n}(\Pi', \mathbf{U}, M)) + d_{SD}(\Pi(\mathbf{P}), \Pi'(\mathbf{P})) && \text{(Triangle ineq.)} \\
&\quad + d_{SD}(\mathrm{Manip}_{m,n}(\Pi', \mathbf{U}, M), \mathrm{Manip}_{m,n}(\Pi, \mathbf{U}, M_\delta)) \\
&\le d_{SD}(\Pi'(\mathbf{P}), \mathrm{Manip}_{m,n}(\Pi', \mathbf{U}, M)) + n\delta && \text{(From Lemma A.2)} \\
&\quad + d_{SD}(\mathrm{Manip}_{m,n}(\Pi', \mathbf{U}, M), \mathrm{Manip}_{m,n}(\Pi, \mathbf{U}, M_\delta)) \\
&\le d_{SD}(\Pi'(\mathbf{P}), \mathrm{Manip}_{m,n}(\Pi', \mathbf{U}, M)) + 2n\delta && \text{(From Lemma A.2)} \\
&\le \gamma + 2n\delta
\end{aligned}$$

This completes the proof. □

It remains to prove Lemma A.2.

Proof of Lemma A.2. For every $x \in [d]$, Lemma 3.1 implies that there are four distributions $R_x^{(+1)}, R_x^{(-1)}, R_x^{\perp}, R_x^{\top}$ such that we can express $R(x)$ and $R(1)$ as the following mixtures:

$$\begin{aligned}
R(x) &= \frac{e^\varepsilon}{e^\varepsilon + 1} \cdot (1-\delta) \cdot R_x^{(+1)} + \frac{1}{e^\varepsilon + 1} \cdot (1-\delta) \cdot R_x^{(-1)} + \delta \cdot R_x^{\perp} \\
R(1) &= \frac{1}{e^\varepsilon + 1} \cdot (1-\delta) \cdot R_x^{(+1)} + \frac{e^\varepsilon}{e^\varepsilon + 1} \cdot (1-\delta) \cdot R_x^{(-1)} + \delta \cdot R_x^{\top}
\end{aligned}$$

Now define the new distribution $R'(x) := \frac{e^\varepsilon}{e^\varepsilon+1} \cdot (1-\delta) \cdot R_x^{(+1)} + \frac{1}{e^\varepsilon+1} \cdot (1-\delta) \cdot R_x^{(-1)} + \delta \cdot R_x^{\top}$. It is clear that the statistical distance between $R(x)$ and $R'(x)$ is at most δ.

Define the mixture distributions $R_{x,+1} := \frac{e^\varepsilon}{e^\varepsilon+1} \cdot R_x^{(+1)} + \frac{1}{e^\varepsilon+1} \cdot R_x^{(-1)}$ and $R_{x,-1} := \frac{1}{e^\varepsilon+1} \cdot R_x^{(+1)} + \frac{e^\varepsilon}{e^\varepsilon+1} \cdot R_x^{(-1)}$. They satisfy the following:

$$\forall y \in \mathcal{Y}\ e^{-\varepsilon} \cdot \mathbb{P}[R_{x,-1} = y] \le \mathbb{P}[R_{x,+1} = y] \le e^\varepsilon \cdot \mathbb{P}[R_{x,-1} = y].$$

Observe that we can sample from $R'(x)$ by post-processing $R_{x,+1}$. Likewise, we can sample from $R(1)$ by using the same post-processing on $R_{x,-1}$. Hence,

$$\forall y \in \mathcal{Y}\ e^{-\varepsilon} \cdot \mathbb{P}[R(1) = y] \le \mathbb{P}\big[R'(x) = y\big] \le e^\varepsilon \cdot \mathbb{P}[R(1) = y].$$

By repeating the same argument for every $x' \in [d]$,

$$\forall x \sim x' \in [d]\ \forall y \in \mathcal{Y}\ e^{-2\varepsilon} \cdot \mathbb{P}\big[R'(x') = y\big] \le \mathbb{P}\big[R'(x) = y\big] \le e^{2\varepsilon} \cdot \mathbb{P}\big[R'(x') = y\big]$$

which immediately implies R' satisfies 2ε-privacy. □

A.2. **Proof of Lemma 4.7.** We will argue that our attack (Algorithm 3) selects a powerful attack against ε-locally private protocols with probability $\ge 2/3$. We first recall notation. For any integer $d > 2$ and algorithm $R : [d] \to \mathcal{Y}$, let $R(\mathbf{U})$ denote the distribution over $\mathcal{Y}$ induced by sampling $\hat{x}$ from the uniform distribution over $[d]$ and then sampling a message from $R(\hat{x})$. For any set $H \subset [d]$, let $R(\mathbf{U}_H)$ denote the distribution over $\mathcal{Y}$ induced by sampling $\hat{x}$ from the uniform distribution over H and then executing $R(\hat{x})$. In this notation, $Q_{H,R}$ is the algorithm which samples from $R(\mathbf{U}_H)$ when given $+1$, but samples from $R(\mathbf{U}_{\overline{H}})$ when given -1.

Our objective is to prove the following, from which Lemma 4.7 is immediate:

Lemma A.3. *Fix any $\delta \in (0,1)$ and any ε-locally private protocol $\Pi = (\vec{R}, A)$ where $d > 4(e^\varepsilon - 1)^2 \ln(24e^\varepsilon n/\delta)$. Suppose H is sampled uniformly from subsets of $[d]$ with size $d/2$. The following holds with probability $> 2/3$ over the randomness of H: all randomizers $\{Q_{H,R_i}\}_{i\in[n]}$ specified by Algorithm 2 satisfy (ε', δ)-privacy, where*

$$\varepsilon' = (e^\varepsilon - 1)\sqrt{\frac{16}{d}\ln\frac{24e^\varepsilon n}{\delta}}$$

The key to the analysis is to argue that, for a uniformly random H, the distributions of $R(\mathbf{U}_H)$ and $R(\mathbf{U})$ are close together. To this end, we introduce the following definition:

Definition A.4 (Leaky Messages)**.** For any $H \subset [d]$ with size $d/2$ and any local randomizer $R : [d] \to \mathcal{Y}$, a message $y \in \mathcal{Y}$ is *v-leaky with respect to H, R* when

$$\frac{\mathbb{P}[R(\mathbf{U}_H) = y]}{\mathbb{P}[R(\mathbf{U}) = y]} \notin [e^{-v}, e^v]$$

Next we show that when y is some fixed message and H is uniformly random, y is not likely to be leaky with respect to H, R.

Claim A.5. Fix any $\varepsilon > 0$, $\beta \in (0,1)$, $d > 4(e^\varepsilon - 1)^2 \ln\frac{2}{\beta}$, and any ε-private randomizer $R : [d] \to \mathcal{Y}$. For any message set $y \in \mathcal{Y}$, if H is chosen uniformly from subsets of $[d]$ with size $d/2$, then with probability at least $1 - \beta$, y is not $(e^\varepsilon - 1)\sqrt{\frac{4}{d}\ln\frac{2}{\beta}}$-leaky with respect to H, R.

Proof. Observe that

$$\begin{aligned}
\mathbb{P}[R(\mathbf{U}) = y] &= \sum_{j=1}^{d} \mathbb{P}[R(j) = y] \cdot \mathop{\mathbb{P}}_{x\sim\mathbf{U}}[x = j] \\
&= \sum_{j=1}^{d} \mathbb{P}[R(j) = y] \cdot \frac{1}{d} && \text{(defn. of } \mathbf{U})
\end{aligned}$$

Also observe that, for any choice of H,

$$\mathbb{P}[R(\mathbf{U}_H) = y] = \sum_{i=1}^{d/2} \mathbb{P}[R(h_i) = y] \cdot \frac{2}{d} \tag{A.1}$$

Due to differential privacy, each term in (A.1) has maximum value $e^\varepsilon \min_j \mathbb{P}[R(j) = y]$. We use the following version of Hoeffding's inequality to bound the summation when H is uniformly random.

Lemma A.6 ((Hoeffding, 1963))**.** *Given a set $\vec{p} = \{p_1, \ldots, p_N\} \in \mathbb{R}^N$ such that $p_i \in (c, c')$, if the subset $\vec{x} = \{x_1, \ldots, x_n\}$ is constructed by uniformly sampling without replacement from $\vec{p}$, then*

$$\mathbb{P}\left[\frac{1}{n}\sum_{i=1}^{n} x_i \le \frac{1}{N}\sum_{i=1}^{N} p_i + (c' - c)\cdot\sqrt{\frac{1}{2n}\log\frac{1}{\beta}}\right] \ge 1 - \beta$$

Hence, the following is true with probability $1 - \beta/2$ over the random choice of H:

$$\begin{aligned}
(\text{A.1}) &\leq \frac{1}{d}\sum_{j=1}^{d} \mathbb{P}[R(j) = y] + \left((e^\varepsilon - 1)\min_j \mathbb{P}[R(j) = y]\right) \cdot \sqrt{\frac{1}{d}\ln\frac{2}{\beta}} \\
&= \mathbb{P}[R(\mathbf{U}) = y] + \left((e^\varepsilon - 1)\min_j \mathbb{P}[R(j) = y]\right) \cdot \sqrt{\frac{1}{d}\ln\frac{2}{\beta}} \\
&\leq \left(1 + (e^\varepsilon - 1)\cdot\sqrt{\frac{1}{d}\ln\frac{2}{\beta}}\right) \cdot \mathbb{P}[R(\mathbf{U}) = y] \\
&\leq \exp\left((e^\varepsilon - 1)\cdot\sqrt{\frac{1}{d}\ln\frac{2}{\beta}}\right) \cdot \mathbb{P}[R(\mathbf{U}) = y]
\end{aligned}$$

By a symmetric version of Lemma A.6, the following holds with probability $1 - \beta/2$:

$$\begin{aligned}
(\text{A.1}) &\geq \frac{1}{d}\sum_{j=1}^{d} \mathbb{P}[R(j) = y] + \left((e^\varepsilon - 1)\min_j \mathbb{P}[R(j) = y]\right) \cdot \sqrt{\frac{1}{d}\ln\frac{2}{\beta}} \\
&= \mathbb{P}[R(\mathbf{U}) = y] - \left((e^\varepsilon - 1)\min_j \mathbb{P}[R(j) = y]\right)\sqrt{\frac{1}{d}\ln\frac{2}{\beta}} \\
&\geq \left(1 - (e^\varepsilon - 1)\cdot\sqrt{\frac{1}{d}\ln\frac{2}{\beta}}\right) \cdot \mathbb{P}[R(\mathbf{U}) = y] \\
&\geq \exp\left(-(e^\varepsilon - 1)\cdot\sqrt{\frac{4}{d}\ln\frac{2}{\beta}}\right) \cdot \mathbb{P}[R(\mathbf{U}) = y]
\end{aligned}$$

The final inequality follows from the condition that $d > 4(e^\varepsilon - 1)^2 \ln\frac{2}{\beta}$. Our claim follows from a union bound. □

If the set H is drawn uniformly and d is sufficiently large, then for most users, we argue that the probability that $R_i(\mathbf{U})$ reports a leaky message is small.

Claim A.7. Fix any $\varepsilon > 0$, $\beta \in (0, 1)$, $d > 4(e^\varepsilon - 1)^2 \ln\frac{2}{\beta}$, and any vector of ε-private randomizers $\vec{R} = (R_1 \dots R_n)$. Suppose H is sampled uniformly from subsets of $[d]$ with size $d/2$. The following holds with probability $\geq 5/6$ over the randomness of H: for all $i \in [n]$, the probability that $R_i(\mathbf{U})$ is $(e^\varepsilon - 1)\sqrt{\frac{4}{d}\ln\frac{2}{\beta}}$-leaky with respect to H, R_i is at most $6\beta n$.

Proof. Let $\mathit{Leak}(v, H, R)$ be the set of all messages y where y is v-leaky with respect to H, R. By Markov's inequality, the following holds for any $i \in [n]$ with probability $\geq 1 - 1/6n$ over the randomness of H:

$$\begin{aligned}
&\mathbb{P}\left[R_i(\mathbf{U}) \in \mathit{Leak}\left((e^\varepsilon - 1)\sqrt{\frac{4}{d}\ln\frac{2}{\beta}}, H, R_i\right)\right] \\
&\leq 6n \cdot \mathop{\mathbb{E}}_{H}[\mathbb{P}[R_i(\mathbf{U}) \in \mathit{Leak}(\dots, H, R_i)]]
\end{aligned}$$

where we use $\dots$ to suppress the v term. Once we bound the expectation by β, our claim follows by a union bound over the n randomizers. Below, we use $\binom{[d]}{d/2}$ as shorthand for the

subsets of $[d]$ with size $d/2$.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{H}[\mathbb{P}[R_i(\mathbf{U}) \in \mathit{Leak}(\ldots, H, R_i)]] \\
&= \sum_{H \in \binom{[d]}{d/2}} \binom{d}{d/2}^{-1} \cdot \mathbb{P}[R_i(\mathbf{U}) \in \mathit{Leak}(\ldots, H, R_i)] \\
&= \sum_{H \in \binom{[d]}{d/2}} \binom{d}{d/2}^{-1} \cdot \sum_{y \in \mathit{Leak}(\ldots, H, R_i)} \mathbb{P}[R_i(\mathbf{U}) = y] \\
&= \sum_{y \in \mathcal{Y}} \mathbb{P}[R_i(\mathbf{U}) = y] \cdot \sum_{H \in \binom{[d]}{d/2}} \binom{d}{d/2}^{-1} \cdot \mathbb{1}_{\{y \in \mathit{Leak}(\ldots, H, R_i)\}} \\
&\leq \sum_{y \in \mathcal{Y}} \beta \cdot \mathbb{P}[R_i(\mathbf{U}) = y] && \text{(Claim A.5)} \\
&= \beta
\end{aligned}$$

This concludes the proof. □

Claim A.7 is a bound on the probability that some $R_i(\mathbf{U})$ is leaky. Because $\vec{R}$ satisfies differential privacy, it implies a bound on the probability that some $R_i(\mathbf{U}_H)$ is leaky.

Corollary A.8. *Fix any $\varepsilon > 0$, $\beta \in (0,1)$, $d > 4(e^\varepsilon - 1)^2 \ln \frac{2}{\beta}$, and any vector of ε-private randomizers $\vec{R} = (R_1 \ldots R_n)$. Suppose H is sampled uniformly from subsets of $[d]$ with size $d/2$. The following holds with probability $\geq 5/6$ over the randomness of H: for all $i \in [n]$, the probability that $R_i(\mathbf{U}_H)$ is $(e^\varepsilon - 1)\sqrt{\frac{4}{d} \ln \frac{2}{\beta}}$-leaky with respect to H, R_i is at most $e^\varepsilon \cdot 6\beta n$.*

Again recall that Q_{H,R_i} reports either a sample from $R_i(\mathbf{U}_H)$ or from $R_i(\mathbf{U}_{\overline{H}})$. Having bounded the probability that either sample is leaky, we can now argue that each Q_{H,R_i} satisfies a stronger level of differential privacy than R_i (Lemma A.3).

Proof of Lemma A.3. Define $v := \varepsilon'/2 = (e^\varepsilon - 1)\sqrt{\frac{4}{d} \ln \frac{24e^\varepsilon n}{\delta}}$. By Corollary A.8, the following holds with probability $> 5/6$ for every $Y \subseteq \mathcal{Y}$:

$$\begin{aligned}
&\mathbb{P}[R_i(\mathbf{U}_H) \in Y] \\
&= \mathbb{P}[R_i(\mathbf{U}_H) \in Y - \mathit{Leak}(v, H, R_i)] \\
&\quad + \mathbb{P}[R_i(\mathbf{U}_H) \in Y \cap \mathit{Leak}(v, H, R_i)] \\
&\leq \mathbb{P}[R_i(\mathbf{U}_H) \in Y - \mathit{Leak}(v, H, R_i)] + \delta/2 \\
&\leq e^v \cdot \mathbb{P}[R_i(\mathbf{U}) \in Y] + \delta/2
\end{aligned}$$

We now justify the final inequality:

$$\begin{aligned}
&\mathbb{P}[R_i(\mathbf{U}_H) \in Y - \mathit{Leak}(v, H, R_i)] \\
&= \sum_{y \in Y - \mathit{Leak}(v,H,R_i)} \mathbb{P}[R_i(\mathbf{U}_H) = y] \\
&\le e^v \sum_{y \in Y - \mathit{Leak}(v,H,R_i)} \mathbb{P}[R_i(\mathbf{U}) = y] + \sum_{y \in Y - \mathit{Leak}(v,H,R_i)} w \qquad \text{(defn. A.4)} \\
&\le e^v \mathbb{P}[R_i(\mathbf{U}) \in Y - \mathit{Leak}(v, H, R_i)] + |\mathcal{Y}| \cdot w
\end{aligned}$$

By symmetric steps and Claim A.7,

$$\mathbb{P}[R_i(\mathbf{U}) \in Y] \le e^v \cdot \mathbb{P}[R_i(\mathbf{U}_H) \in Y] + \delta/2$$

We take identical steps to show that the following holds with probability $> 5/6$ as well:

$$\begin{aligned}
\mathbb{P}\big[R_i(\mathbf{U}_{\overline{H}}) \in Y\big] &\le e^v \cdot \mathbb{P}[R_i(\mathbf{U}) \in Y] + \delta \\
\mathbb{P}[R_i(\mathbf{U}) \in Y] &\le e^v \cdot \mathbb{P}\big[R_i(\mathbf{U}_{\overline{H}}) \in Y\big] + \delta
\end{aligned}$$

From a union bound, the following holds with probability $> 2/3$:

$$\begin{aligned}
\mathbb{P}\big[R_i(\mathbf{U}_{\overline{H}}) \in Y\big] &\le e^{2v} \cdot \mathbb{P}[R_i(\mathbf{U}_H) \in Y] + \delta \\
\mathbb{P}[R_i(\mathbf{U}_H) \in Y] &\le e^{2v} \cdot \mathbb{P}\big[R_i(\mathbf{U}_{\overline{H}}) \in Y\big] + \delta
\end{aligned}$$

Because Q_{H,R_i} samples from $R_i(\mathbf{U}_H)$ on input $+1$ and from $R_i(\mathbf{U}_{\overline{H}})$ on input -1, it satisfies $(2v, \delta)$-privacy. Substitution of $v = \varepsilon'/2$ concludes the proof. □

A.3. **Alternate version of Lemma 4.7.** As claimed in Section 4, we can remove the $\ln n$ factor in the special case where there are $O(1)$ unique randomizers and $O(1)$ possible messages. We will require the following corollary:

Corollary A.9. *Fix any vector of ε-private randomizers $\vec{R} = (R_1, \ldots, R_n)$ where the number of unique randomizers is κ_R, the message universe has size $|\mathcal{Y}|$, and $d > 4(e^\varepsilon - 1)^2 \ln(12\kappa_R|\mathcal{Y}|)$. Sample H uniformly at random over subsets of $[d]$ with size $d/2$. The following is true with probability $\ge 5/6$ over the randomness of H: $\forall y \in \mathcal{Y}\ \forall i \in [n]$ y is not $(e^\varepsilon - 1)\sqrt{\frac{4}{d} \ln 12\kappa_R|\mathcal{Y}|}$-leaky w.r.t. H, R_i*

The statement is immediate from Claim A.5 and a union bound. We now state and prove the alternate lemma:

Lemma A.10. *Fix any ε-locally private protocol $\Pi = (\vec{R}, A)$ where the number of unique randomizers is κ_R, the message universe has size $|\mathcal{Y}|$, and $d > 4(e^\varepsilon - 1)^2 \ln(12|\mathcal{Y}| \cdot \kappa_R)$. Suppose H is sampled uniformly from subsets of $[d]$ with size $d/2$. The following holds with probability $> 2/3$ over the randomness of H: all randomizers $\{Q_{H,R_i}\}_{i \in [n]}$ specified by Algorithm 2 satisfy ε'-privacy, where $\varepsilon' = (e^\varepsilon - 1)\sqrt{\frac{16}{d} \ln(12\kappa_R|\mathcal{Y}|)}$. When $\varepsilon = O(1)$, $\kappa_R = O(1)$, and $|\mathcal{Y}| = O(1)$, this parameter is $O(\varepsilon/\sqrt{d})$.*

Proof. Recall the definition of Q_{H,R_i}: on input $+1$, it samples from $R_i(\mathbf{U}_H)$ and, on input -1, it samples from $R_i(\mathbf{U}_{\overline{H}})$. We will bound how leaky any message set $Y \subseteq \mathcal{Y}$ can be via Corollary A.9: with probability $\geq 5/6$, the following holds for all $i \in [n]$:

$$\begin{aligned}
&\mathbb{P}[R_i(\mathbf{U}_H) \in Y] \\
&= \sum_{y=y} \mathbb{P}[R_i(\mathbf{U}_H) = y] \\
&\leq \sum_{y=y} \exp\left((e^\varepsilon - 1) \cdot \sqrt{\frac{4}{d} \ln 12\kappa_R |\mathcal{Y}|}\right) \cdot \mathbb{P}[R_i(\mathbf{U}) = y] \\
&\leq \exp\left((e^\varepsilon - 1) \cdot \sqrt{\frac{4}{d} \ln 12\kappa_R |\mathcal{Y}|}\right) \cdot \mathbb{P}[R_i(\mathbf{U}) \in Y]
\end{aligned}$$

and again with probability $\geq 5/6$

$$\begin{aligned}
&\mathbb{P}[R_i(\mathbf{U}) \in Y] \\
&\leq \exp\left((e^\varepsilon - 1) \cdot \sqrt{\frac{4}{d} \ln 12\kappa_R |\mathcal{Y}|}\right) \cdot \mathbb{P}\big[R_i(\mathbf{U}_{\overline{H}}) \in Y\big]
\end{aligned}$$

From a union bound, we can conclude that

$$\begin{aligned}
&\mathbb{P}[R_i(\mathbf{U}_H) \in Y] \\
&\leq \exp\left((e^\varepsilon - 1) \cdot \sqrt{\frac{16}{d} \ln 12\kappa_R |\mathcal{Y}|}\right) \cdot \mathbb{P}\big[R_i(\mathbf{U}_{\overline{H}}) \in Y\big]
\end{aligned}$$

with probability $\geq 2/3$. By symmetric steps, we can obtain the inequality where the positions of $\mathbb{P}\big[R_i(\mathbf{U}_{\overline{H}}) \in Y\big]$ and $\mathbb{P}[R_i(\mathbf{U}_H) \in Y]$ are swapped. □

Appendix B. Construction and Analysis of Protocols from Section 5

In Section 5, we sketched some protocols and bounded the effect of manipulation attacks on them. In this Appendix, we provide more details for these protocols.

B.1. Construction and Analysis of $\texttt{EST}\infty$.

The protocol $\texttt{EST}\infty_{n,d,\varepsilon}$ is designed to estimate d counting queries; as suggested by the name, it attempts to minimize error in ℓ_∞ norm. It consists of the n randomizers $(R^{\texttt{EST}\infty}_{n,d,\varepsilon,i})_{i\in[n]}$ and the aggregator $A^{\texttt{EST}\infty}_{n,d,\varepsilon}$; see Algorithms 4 and 5 for the pseudocode. A public partition of $[n]$ into d groups, denoted π, is drawn uniformly at random.

An important subroutine is described by (B.1). It samples from ± 1 in such a way that the mean is equal to the jth coordinate of user data x_i.

$$\mathit{Encode}_\infty(x_i, j) := \begin{cases} +1 & \text{with probability } \frac{1}{2} + \frac{x_{i,j}}{2} \\ -1 & \text{with probability } \frac{1}{2} - \frac{x_{i,j}}{2} \end{cases} \tag{B.1}$$

The following statement is a version of Theorem 5.2 that allows for arbitrary failure probability.

Algorithm 4: $R^{\mathtt{EST}\infty}_{n,d,\varepsilon,i}(x_i, \pi)$

Parameters: $n, d \in \mathbb{Z}^+$, $\varepsilon > 0$, $i \in [n]$
Input: $x_i \in B^d_\infty$; π, a public partition of $[n]$ into d groups
Output: $y_i \in \{-\frac{e^\varepsilon+1}{e^\varepsilon-1}, \frac{e^\varepsilon+1}{e^\varepsilon-1}\}$

$g(i) \leftarrow$ the group i belongs to in π
$x'_i \leftarrow \mathit{Encode}_\infty(x_i, g(i))$
$y_i \leftarrow R^{\mathtt{RR}}(x'_i)$
Return y_i

Algorithm 5: $A^{\mathtt{EST}\infty}_{n,d,\varepsilon}(y_1, \ldots, y_n, \pi)$

Parameters: $n, d \in \mathbb{Z}^+, \varepsilon > 0$
Input: $\vec{y} \in \{-\frac{e^\varepsilon+1}{e^\varepsilon-1}, \frac{e^\varepsilon+1}{e^\varepsilon-1}\}^n$; π, a public partition of $[n]$ into d groups
Output: $\vec{z} \in \mathbb{R}^d$

For $g : 1 \to d$
  $\pi(g) \leftarrow$ the gth group of $[n]$
  $z_g \leftarrow \frac{d}{n} \sum_{i \in \pi(g)} y_i$
Return $\vec{z}$

Theorem B.1. *For any $\beta \in (0,1)$, there is a constant c such that, for any $\varepsilon > 0$, any positive integers $m \le n$, any $x_1, \ldots, x_n \in B^d_\infty$, and any attacker M* oblivious to public randomness*:*

$$\mathbb{P}\left[\left\|\mathrm{Manip}(\mathit{EST}\infty_{n,d,\varepsilon}, \vec{x}, M) - \frac{1}{n}\sum_{i=1}^n x_i\right\|_\infty < c \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \left(\sqrt{\frac{d}{n}\log\frac{d}{\beta}} + \frac{m}{n}\right)\right] \ge 1-\beta$$

To prove the theorem, we bound the error introduced by each source of randomness. We first consider the difference between the underlying mean and the mean in a partition given by π. Hoeffding's inequality and a union bound yields the following claim:

Claim B.2. Fix any $x_1, \ldots, x_n \in B^d_\infty$. There is a constant c such that, when π is a uniformly random partition of $[n]$ into d groups $\pi(1), \ldots, \pi(d)$,

$$\mathbb{P}\left[\forall g \in [d] \left|\frac{1}{n}\sum_{i=1}^n x_{i,g} - \frac{d}{n}\sum_{i \in \pi(g)} x_{i,g}\right| < c \cdot \sqrt{\frac{d}{n}\ln\frac{d}{\beta}}\right] \ge 1-\beta$$

The protocol executes Encode_∞ on all $x_{i,g(i)}$. Here, we use the Hoeffding inequality again to bound the error introduced by this encoding:

Claim B.3. Fix any $x_1, \ldots, x_n \in B^d_\infty$ and any partition π of $[n]$ into d groups $\pi(1), \ldots, \pi(d)$. Suppose, for every $g \in [d]$, we execute $x'_i \leftarrow \mathit{Encode}_\infty(x_i, g)$ for each user $i \in \pi(g)$.

$$\mathbb{P}\left[\forall g \in [d] \left|\frac{d}{n}\sum_{i \in \pi(g)} x_{i,g} - \frac{d}{n}\sum_{i \in \pi(g)} x'_i\right| < c \cdot \sqrt{\frac{d}{n}\ln\frac{d}{\beta}}\right] \ge 1-\beta$$

If an attacker chooses the set of corrupt users C independently of π, we use a Chernoff bound to bound the number of corruptions in any group:

Claim B.4. Fix any $n, d \in \mathbb{Z}^+$ and any set of corrupted users $C \subset [n]$ where $|C| = m$. There is a constant c such that, when π is a uniformly random partition of $[n]$ into d groups $\pi(1), \ldots, \pi(d)$,

$$\mathbb{P}\left[\forall g \in [d]\ |C \cap \pi(g)| < \frac{m}{d} + c \cdot \sqrt{\frac{m}{d} \ln \frac{d}{\beta}}\right] \geq 1 - \beta$$

When we apply randomized response to data encoded by $Encode_\infty$, we can obtain our third bound on error immediately from Theorem 5.1:

Claim B.5. For any $m' \leq n/d$, any $\vec{x}' \in \{\pm 1\}^{n/d}$ and any attacker M, $\texttt{RR}_{\varepsilon/2, n/d}$ has the following guarantee on estimation error after playing the $(m', n/d)$-manipulation game:

$$\mathbb{P}\left[\left|\text{Manip}_{m', n/d}(\texttt{RR}_{\varepsilon, n/d}, \vec{x}', M) - \frac{d}{n}\sum_{i=1}^{n/d} x'_i\right| < \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \left(\sqrt{\frac{2d}{n} \ln \frac{2d}{\beta}} + \frac{2dm'}{n}\right)\right] \geq 1 - \beta/d$$

Theorem 5.2 follows from union bounds over Claims B.2, B.3, B.4, and B.5 (we substitute m' in Claim B.5 with the upper bound in Claim B.4).

B.2. **Construction and Analysis of `EST1`.** The protocol $\texttt{EST1}_{n,d,\varepsilon}$ takes in user data from the unit ball and performs private mean estimation. The protocol consists of the n randomizers $(R^{\texttt{EST1}}_{n,d,\varepsilon,i})_{i \in [n]}$ and the aggregator $A^{\texttt{EST1}}_{n,d,\varepsilon}$ (see Algorithms 6 and 8, respectively). Each user i is associated with a public vector $\vec{s}_i \in \{\pm 1\}^{2d+1}$ which is sampled uniformly and independently at random.

Algorithm 6: $R^{\texttt{EST1}}_{n,d,\varepsilon,i}(x_i, \vec{s}_i)$

Parameters: $n, d \in \mathbb{Z}^+, \varepsilon > 0, i \in [n]$
Input: $x \in B_1^d$ and $\vec{s}_i \in \{\pm 1\}^{2d+1}$
Output: $y \in \{\pm \frac{e^\varepsilon + 1}{e^\varepsilon - 1}\}$

$x'_i \leftarrow Encode_1^d(x_i)$ `/* See Algorithm 7 */`
$y_i \leftarrow R^{\texttt{RR}}_\varepsilon(s_{i,x'})$
Return y_i

We give bounds on both ℓ_∞ and ℓ_1 error in separate subsections. We focus on the error absent manipulation: the bounds on `HST`'s manipulation error we gave in Section 5 also hold for `EST1`. This is because `EST1` is essentially the composition of `HST` and a pre-processing step that converts points in B_1^d to members of $[d]$ (see Algorithm 7).

B.2.1. *Error in ℓ_∞.* In this subsection, we bound the maximum error of `EST1` along any dimension.

Theorem B.6. *There is a constant c such that, for any $\beta \in (0, 1)$, any $\varepsilon > 0$, any positive integers n, d, and any $\vec{x} = (x_1, \ldots, x_n) \in [d]^n$, with probability $\geq 1 - \beta$, we have*

$$\left\|\mathit{EST1}_{n,d,\varepsilon}(\vec{x}) - \frac{1}{n}\sum_{i=1}^{n} x_i\right\|_\infty \leq c \cdot \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \sqrt{\frac{1}{n} \ln \frac{d}{\beta}}$$

Algorithm 7: $\mathit{Encode}_1^d(x)$

Parameters: $d \in \mathbb{Z}^+$
Input: $x \in B_1^d$
Output: $x' \in [2d+1]$

For $j \in [d]$
  If $x_j > 0$
    $p_{2j-1} \leftarrow x_j$
    $p_{2j} \leftarrow 0$
  Else
    $p_{2j-1} \leftarrow 0$
    $p_{2j} \leftarrow -x_j$
$p_{2d+1} \leftarrow 1 - \|x\|_1$
Sample x' from the distribution over $[2d+1]$ such that $\mathbb{P}[x' = k] = p_k$
Return x'

Algorithm 8: $A_{n,d}^{\mathtt{EST1}}(y_1, \vec{s}_1, \ldots, y_n, \vec{s}_n)$

Parameters: $n, d \in \mathbb{Z}^+$
Input: $y_i \in \{\pm\frac{e^\varepsilon+1}{e^\varepsilon-1}\}$ and $\vec{s}_i \in \{\pm 1\}^{2d+1}$ for every $i \in [n]$
Output: $\hat{\mu} \in \mathbb{R}^d$

For $j' \in [2d+1]$
  $z_{j'} \leftarrow \frac{1}{n}\sum_{i=1}^n y_i s_{i,j'}$
For $j \in [d]$
  $\hat{\mu}_j \leftarrow z_{2j-1} - z_{2j}$
$\hat{\mu} \leftarrow (\hat{\mu}_1, \ldots, \hat{\mu}_d)$
Return $\hat{\mu}$

To prove the theorem, we bound the error introduced by Encode_∞^d and by $R_\varepsilon^{\mathtt{RR}}$ separately. We use shorthand $\mathit{freq}(j, \vec{x}) := \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{x_i = j\}}$ and $\mathit{freq}(\vec{x}) := (\mathit{freq}(1, \vec{x}), \ldots, \mathit{freq}(d, \vec{x}))$.

Claim B.7. There is a constant c such that for any positive integers n, d and any $x_1, \ldots, x_n \in B_1^d$, if we sample $x_i' \leftarrow \mathit{Encode}_1^d(x_i)$ for each user i, then

$$\mathbb{P}\left[\max_{j\in[d]}\left|(\mathit{freq}(2j-1, \vec{x}\,') - \mathit{freq}(2j, \vec{x}\,')) - \frac{1}{n}\sum_{i=1}^n x_{i,j}\right| \le c \cdot \sqrt{\frac{1}{n}\ln\frac{d}{\beta}}\right] \ge 1 - \beta$$

Proof. Consider any user data $x_i \in B_1^d$ and any coordinate $j \in [d]$. Without loss of generality, we will assume that $x_{i,j} > 0$. By construction, $\mathbb{P}[x_i' = 2j-1] = x_i$ and $\mathbb{P}[x_i' = 2j] = 0$. Hence,

$$\mathbb{E}\Big[\mathbb{1}_{\{x_i'=2j-1\}} - \mathbb{1}_{\{x_i'=2j\}}\Big] = x_{i,j}$$

$$\mathbb{E}\left[\sum_{i=1}^n \mathbb{1}_{\{x_i'=2j-1\}} - \sum_{i=1}^n \mathbb{1}_{\{x_i'=2j\}}\right] = \sum_{i=1}^n x_{i,j}$$

$$\mathbb{E}\big[\mathit{freq}(2j-1, \vec{x}\,') - \mathit{freq}(2j, \vec{x}\,')\big] = \frac{1}{n}\sum_{i=1}^n x_{i,j}$$

The random variable $\mathbb{1}_{\{x'_i=2j-1\}} - \mathbb{1}_{\{x'_i=2j\}} - x_{i,j}$ ranges from -2 to $+2$. By a Hoeffding bound, the following holds with probability $\geq 1 - \beta/d$.

$$\left| \mathit{freq}(2j-1, \vec{x}\,') - \mathit{freq}(2j, \vec{x}\,') - \frac{1}{n}\sum_{i=1}^{n} x_{i,j} \right| < \sqrt{\frac{8}{n} \ln \frac{2d}{\beta}}$$

A union bound over all $j \in [d]$ completes the proof. □

Claim B.8. There is a constant c such that for any positive integers n, d and any $x'_1, \ldots, x'_n \in [2d+1]$, if we sample $y_i \leftarrow R^{\mathtt{RR}}_{\varepsilon}(s_{i,x'_i})$ for each user i and compute $z_{j'} = \frac{1}{n}\sum_{i=1}^{n} y_i s_{i,j'}$ for each $j' \in [2d+1]$, then

$$\mathbb{P}\left[\left\| \vec{z} - \mathit{freq}(\vec{x}\,') \right\|_\infty \leq c \cdot \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \sqrt{\frac{1}{n} \ln \frac{d}{\beta}} \right] \geq 1 - \beta$$

Proof. To prove this claim, we fix a value $j' \in [2d+1]$ and argue that the estimate of $\mathit{freq}(j', \vec{x}\,')$ has error $c \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \sqrt{\frac{1}{n}\ln\frac{d}{\beta}}$ with probability $1 - \beta/(2d+1)$. A union bound over all $j' \in [2d+1]$ will complete the proof.

Recall that all y_i have magnitude $\frac{e^\varepsilon+1}{e^\varepsilon-1}$. By a Hoeffding bound, the following holds with probability $\geq 1 - \beta/(2d+1)$ (where we use $\vec{s}(j')$ to denote the vector $(s_{1,j'}, \ldots, s_{n,j'})$):

$$\left| z_{j'} - \mathop{\mathbb{E}}_{R^{\mathtt{RR}}, \vec{s}(j')}\left[z_{j'}\right] \right| < \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \sqrt{\frac{2}{n} \ln \frac{2(2d+1)}{\beta}}$$

It remains to show that $z_{j'}$ is unbiased:

$$\begin{aligned}
\mathop{\mathbb{E}}_{R^{\mathtt{RR}}, \vec{s}(j')}\left[z_{j'}\right] &= \frac{1}{n}\sum_{i=1}^{n} \mathop{\mathbb{E}}_{R^{\mathtt{RR}}, \vec{s}(j')}\left[y_i s_{i,j'}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathop{\mathbb{E}}_{\vec{s}(j')}\left[s_{i,j'} \cdot s_{i,x_i}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{x_i = j'\}} = \mathit{freq}(j', \vec{x}\,')
\end{aligned}$$

This concludes the proof. □

We also formalize the argument that m users cannot manipulate the protocol beyond $O(m/\varepsilon n)$:

Theorem B.9. *There is a constant c such that, for any $\beta \in (0,1)$, any $\varepsilon > 0$, any positive integers m, n, d, any attacker M, and any $\vec{x} = (x_1, \ldots, x_n) \in [d]^n$, with probability $\geq 1 - \beta$, we have*

$$\left\| \mathrm{Manip}_{m,n}(\mathit{EST1}_{n,d,\varepsilon}, \vec{x}, M) - \frac{1}{n}\sum_{i=1}^{n} x_i \right\|_\infty \leq c \cdot \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \left(\sqrt{\frac{1}{n}\ln\frac{d}{\beta}} + \frac{m}{n} \right)$$

Proof. As sketched in Section 5, we bound the error from the honest execution separately from the error from the manipulation. Given that Theorem B.6 already bounds the honest

execution, it will suffice to prove that

$$\max_{j\in[2d+1]}\left|\frac{1}{n}\sum_{i\in C}(y_i-\underline{y}_i)s_{i,j'}\right|\le 2\cdot\frac{e^\varepsilon+1}{e^\varepsilon-1}\cdot\frac{m}{n}$$

By construction, both the manipulative messages y_i and the honest messages $\underline{y}_i$ are members of the set $\{\pm\frac{e^\varepsilon+1}{e^\varepsilon-1}\}$. There are m corrupt users in C. Hence, the bound follows. □

B.2.2. *Error in ℓ_1.* In this subsection, we bound the ℓ_1 error of the `EST1` protocol.

Theorem B.10. *There is a constant c such that, for any $\beta\in(0,1)$, any $\varepsilon>0$, any positive integers n,d, and any $\vec{x}=(x_1,\ldots,x_n)\in[d]^n$, with probability $\ge 1-\beta$, we have*

$$\left\|\mathit{EST1}_{n,d,\varepsilon}(\vec{x})-\frac{1}{n}\sum_{i=1}^n x_i\right\|_1\le c\cdot\frac{e^\varepsilon+1}{e^\varepsilon-1}\cdot\sqrt{\frac{d^2}{n}\log\frac{1}{\beta}}$$

To prove the theorem, we bound the error introduced by Encode_1^d and by $R_\varepsilon^{\mathtt{RR}}$ separately. We begin with Encode_1^d.

Claim B.11. There is a constant c such that for any positive integers n,d and any $x_1,\ldots,x_n\in B_1^d$, if we sample $x_i'\leftarrow\mathit{Encode}_1^d(x_i)$ for each user i, then the following holds with probability $\ge 1-\beta$:

$$\sum_{j\in[d]}\left|(\mathit{freq}(2j-1,\vec{x}')-\mathit{freq}(2j,\vec{x}'))-\frac{1}{n}\sum_{i=1}^n x_{i,j}\right|\le c\cdot\sqrt{\frac{d^2}{n}\log\frac{1}{\beta}}\tag{B.2}$$

Proof. For any $i\in[n],j\in[d]$, define the random variable $\mathit{err}(i,j):=\mathbb{1}_{\{x_i'=2j-1\}}-\mathbb{1}_{\{x_i'=2j\}}-x_{i,j}$. Observe that $\mathbb{E}[\mathit{err}(i,j)]=0$ and $|\mathit{err}(i,j)|\le 2$. By Hoeffding's inequality, the quantity $\frac{1}{n}\sum_{i=1}^n\mathit{err}(i,j)$ is subgaussian. Specifically, for all $t>0$,

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^n\mathit{err}(i,j)\right|>t\right]\le 2\exp(-nt^2/8)\tag{B.3}$$

Note that this implies there are constants c_0,c_1 such that

$$\mathbb{E}\left[\left|\frac{1}{n}\sum_{i=1}^n\mathit{err}(i,j)\right|\right]\le c_0\cdot\sqrt{\frac{1}{n}}\tag{B.4}$$

$$\mathrm{Var}\left[\left|\frac{1}{n}\sum_{i=1}^n\mathit{err}(i,j)\right|\right]\le c_1\cdot\frac{1}{n}\tag{B.5}$$

For shorthand, we define $\mathit{err}(j):=\left|\frac{1}{n}\sum_{i=1}^n\mathit{err}(i,j)\right|\le 2$. Observe that the left-hand side of (B.2) is equivalent to $\sum_{j=1}^d\mathit{err}(j)$. From (B.4), (B.4), and a Chernoff bound, the

Algorithm 9: $R^{\texttt{RAPTOR}}_{\varepsilon,g}$, randomizer for uniformity testing

Parameters: *Privacy parameter* ε*; group number* $g \in [G]$
Input: $x \in [d]$; public sets $S_1, \dots, S_G$ where $S_g \subset [d]$ and $|S_g| = d/2$
Output: $y \in \{\pm\frac{e^\varepsilon+1}{e^\varepsilon-1}\}$

$x' \leftarrow \mathbb{1}_\pm[x \in S_g]$
$y \sim R^{\texttt{RR}}_\varepsilon(x')$
Return y

sum tightly concentrated around its expectation:

$$\mathbb{P}\left[\sum_{j=1}^{d} err(j) > d \cdot \mathbb{E}[err(1)] + \sqrt{d\mathrm{Var}[err(1)]\log\frac{1}{\beta}}\right] \le \beta$$

$$\mathbb{P}\left[\sum_{j=1}^{d} err(j) > c_0 \cdot \sqrt{\frac{d^2}{n}} + \sqrt{c_1 \cdot \frac{d}{n}\log\frac{1}{\beta}}\right] \le \beta \qquad \text{(From (B.4) and (B.5))}$$

This concludes the proof. □

Now we bound the error due to $R^{\texttt{RR}}_\varepsilon$.

Claim B.12. There is a constant c such that for any positive integers n, d and any $x'_1, \dots, x'_n \in [2d+1]$, if we sample $y_i \leftarrow R^{\texttt{RR}}_\varepsilon(s_{i,x'_i})$ for each user i and compute $z_{j'} = \frac{1}{n}\sum_{i=1}^n y_i s_{i,j'}$ for each $j' \in [2d+1]$, then the following holds with probability $\ge 1-\beta$

$$\sum_{j' \in [2d+1]} \left|z_{j'} - freq(j', \vec{x}')\right| \le c \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \sqrt{\frac{d^2}{n}\log\frac{1}{\beta}}$$

Proof. Define the random variable $err(i,j') := y_i s_{i,j'} - \mathbb{1}_{\{x'_i = j'\}}$. The same steps taken in the proof of Claim B.11 apply here, except now $|err(i,j')| \le 2 \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1}$. □

B.3. **Construction and Analysis of** `RAPTOR`**.** $\texttt{RAPTOR}_{n,G,\varepsilon}$ is a locally private protocol for uniformity testing derived from (Acharya et al., 2019); recall that the input consists of independent samples from some distribution over $[d]$ and the goal is to identify if the distribution is uniform or α-far from uniform (in ℓ_1 distance). The protocol consists of G randomizers: user i is assigned randomizer $\lceil i/G \rceil$. Public randomness will generate $S_1, \dots, S_G$ each of which are uniformly random subsets of $[d]$ of size $d/2$. If a user runs the g-th randomizer, they will privately report whether or not their data lies in S_g. The aggregator performs a threshold test on each group.

We reproduce the randomizer and aggregator pseudocode in Algorithms 9 and 10. For the sake of this proof, we use $\mathbb{1}_\pm[bool]$ to denote the indicator function that evaluates to $+1$ when *bool* is true and -1 when it false.

We rely on the following technical lemma concerning uniformly random S:

Algorithm 10: $A^{\texttt{RAPTOR}}_{n,G,\varepsilon}$, an aggregation algorithm for uniformity testing

Parameters: *Positive integers* n, G*; privacy parameter* ε
Input: $y_1, \dots y_n \in \{\pm\frac{e^\varepsilon+1}{e^\varepsilon-1}\}^n$; public sets $S_1, \dots, S_G$
Output: The string "Uniform" or the string "Not Uniform"
$\alpha_G := \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \left(\sqrt{\frac{6G}{n} \ln \frac{4G}{\beta}} + \frac{2mG}{n}\right)$

For $g \in [G]$
  $start(g) \leftarrow 1 + (g-1) \cdot n/G$
  $end(g) \leftarrow g \cdot n/G$
  `/* Estimate of probability mass in` S_g `*/`
  $\tilde{p}(S_g) \leftarrow \frac{G}{n} \sum_{i=start(g)}^{end(g)} y_i$
  If $|\tilde{p}(S_g)| > 2\alpha_G$
    Return *"Not uniform"*
Return *"Uniform"*

Lemma B.13 (From (Acharya et al., 2019))**.** *If S is a uniformly random subset of $[d]$ with size $d/2$ and $\|\mathbf{P} - \mathbf{U}\|_1 > \alpha\sqrt{10d}$, then*

$$\mathop{\mathbb{P}}_{S}\left[\left|\frac{1}{2} - \mathop{\mathbb{P}}_{x\sim\mathbf{P}}[x \in S]\right| > \alpha\right] > \frac{1}{477}$$

Corollary B.14. *If S is a uniformly random subset of $[d]$ with size $d/2$ and $\|\mathbf{P} - \mathbf{U}\|_1 > \alpha\sqrt{10d}$, then*

$$\mathop{\mathbb{P}}_{S}\left[\left|\mathop{\mathbb{E}}_{x\sim\mathbf{P}}[\mathbb{1}_{\pm}[x \in S]]\right| > 2\alpha\right] > \frac{1}{477}$$

The following statement is a version of Theorem 5.9 that allows for arbitrary failure probability β.

Theorem B.15. *There is a constant c and a choice of parameter $G = \Theta(\log 1/\beta)$ such that, for any $\varepsilon > 0$, any positive integers $m \le n$, and any attacker M, the following holds with probability $\ge 1 - \beta$*

$$\mathrm{Manip}_{m,n}(\textit{RAPTOR}_{n,G,\varepsilon}, \mathbf{U}, M) = \text{"uniform"}$$

and, when $\|\mathbf{P} - \mathbf{U}\|_1 \ge c \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \left(\sqrt{\frac{dG}{n} \ln \frac{G}{\beta}} + \frac{mG\sqrt{d}}{n}\right)$, the following also holds with probability $\ge 1 - \beta$

$$\mathrm{Manip}_{m,n}(\textit{RAPTOR}_{n,G,\varepsilon}, \mathbf{P}, M) = \text{"not uniform"}$$

Proof. We specify the following undesirable events:

$$E_1 := \exists g \in [G] \left|\mathop{\mathbb{E}}_{x\sim\mathbf{P}}[\mathbb{1}_{\pm}[x \in S]] - \frac{G}{n}\sum_{i=start(g)}^{end(g)} \mathbb{1}_{\pm}[x_i \in S]\right| > \alpha_G$$

$$E_2 := \exists g \in [G] \left|\frac{G}{n}\sum \mathbb{1}_{\pm}[x_i \in S] - \tilde{p}(S_g)\right| > \alpha_G$$

$$E_2 := \forall g \in [G] \left|\mathop{\mathbb{E}}_{x\sim\mathbf{P}}[\mathbb{1}_{\pm}[x \in S]]\right| < 2\alpha_G$$

If $\mathbf{P} = \mathbf{U}$ and neither E_1 nor E_2 have occurred, every $\tilde{p}(S_g)$ is at most $2\alpha_G$. Thus, the output is "Uniform."

Algorithm 11: $\texttt{OneHotHash}_{h,k}(g, x_i)$

```
/* One-hot vector that encodes a hashed value */
```
$x'_i \leftarrow (\underbrace{0, \ldots, 0}_{2k \text{ copies}})$
$h_g(x) \leftarrow 2h(x) - bit(g, x_i)$
$x'_{i,h_g(x)} \leftarrow +1$
Return x'_i

If $\|\mathbf{P} - \mathbf{U}\|_1 \geq \alpha_G \cdot \sqrt{160d}$ and none of E_1, E_2, E_3 have occurred, some $\tilde{p}(S_g)$ has magnitude at least $2\alpha_G$. Thus, the output is "Not uniform."

$\mathbb{P}[E_1] < \beta/3$ follows from a Hoeffding bound. $\mathbb{P}[E_2] < \beta/3$ follows from Theorem 5.1 and a union bound over G plays of $\texttt{RR}_{n/G,\varepsilon}$ in the manipulation game. When $\|\mathbf{P} - \mathbf{U}\|_1 \geq \sqrt{160d} \cdot \alpha_G$ and $G \leftarrow \lceil \ln(2/\beta)/\ln(477/476) \rceil$, $\mathbb{P}[E_3] < \beta/3$ follows from Corollary B.14. A union bound over all three completes the proof. □

Appendix C. Analysis of a Heavy Hitters Protocol

In the heavy hitters problem, each user has data $x_i \in [d]$. The objective is to find a small subset L of the universe that contains every element $j \in [d]$ such that $freq_j(\vec{x}) > \alpha$. Because there are $1/\alpha$ heavy hitters, the size of L should be $O(1/\alpha)$.

We consider the protocol HH described in (Bassily et al., 2017).[4] The protocol $\texttt{HH}_{n,d,k,\varepsilon}$ consists of the n randomizers $(R^{\texttt{HH}}_{n,d,k,\varepsilon,i})_{i\in[n]}$ and the aggregator $A^{\texttt{HH}}_{n,d,k,\varepsilon}$; see Algorithms 12 and 13 for the pseudocode. A public data structure π partitions $[n]$ into $\log_2 d$ groups uniformly at random. We assume the data structure has an implicit order within each group $\pi(1), \ldots, \pi(\log_2 d)$. The public hash function $h : [d] \to [k]$ is drawn uniformly. To facilitate the use of EST1, we also sample vectors $\vec{s}_1, \ldots, \vec{s}_n$ uniformly from $\{\pm 1\}^{2k}$.

Algorithm 12: $R^{\texttt{HH}}_{n,d,k,\varepsilon,i}(x_i, \pi, h, \vec{s}_i)$

Parameters: $n, d, k \in \mathbb{Z}^+$; $\varepsilon > 0$; $i \in [n]$
Input: $x_i \in [d]$; public partition π; public hash $h : [d] \to [k]$; public vector $\vec{s} \in \{\pm 1\}^{2k}$
Output: $y_i \in \{\pm \frac{e^\varepsilon + 1}{e^\varepsilon - 1}\}$

$g(i) \leftarrow$ group that i belongs to in π
$x'_i \leftarrow \texttt{OneHotHash}_{h,k}(g(i), x_i)$
```
/* Contribute to a histogram by reporting the one-hot */
```
$n' \leftarrow n/\log_2 d$
$i' \leftarrow$ index of i in group $g(i)$
$y_i \sim R^{\texttt{EST1}}_{n',2k,\varepsilon,i'}(x'_i, \vec{s}_i)$
Return y_i

[4] In (Bassily et al., 2017), the protocol is called Bitstogram.

Algorithm 13: $A^{\texttt{HH}}_{n,d,k,\varepsilon}(y_1, \vec{s}_1, \ldots, y_n, \vec{s}_n, \pi, h)$

Parameters: $n, d, k \in \mathbb{Z}^+$; $\varepsilon > 0$
Input: $y_i \in \{\pm\frac{e^\varepsilon+1}{e^\varepsilon-1}\}$ and $\vec{s}_i \in \{\pm 1\}^{2k}$ for each $i \in [n]$; public partition π; public hash $h : [d] \to [k]$
Output: $L \subset [d]$ with size k

$n' \leftarrow n/\log_2 d$
For $g : 1 \to \log_2 d$
  `/* Obtain the g-th noisy histogram */`
  $(z_1^{(g)}, \ldots, z_{2k}^{(g)}) \leftarrow A^{\texttt{EST1}}_{n',2k,\varepsilon}(\{y_i, \vec{s}_i\}_{i\in\pi(g)})$

$L \leftarrow \emptyset$
For $v \in [k]$
  `/* Recover the bits of an element in` $[d]$ `that hashes to` v `*/`
  For $g : 1 \to \log_2 d$
    If $z_{2v-1}^{(g)} > z_{2v}^{(g)}$ $bit_v^{(g)} \leftarrow 1$;
    Else $bit_v^{(g)} \leftarrow 0$;
  $L \leftarrow L \cup$ the number represented in binary by $bit_v^{(1)}, \ldots, bit_v^{(\log_2 d)}$
Return L

Theorem C.1. *There is a constant c such that, for any $\varepsilon > 0$, any positive integers $m \le n$, any $\vec{x} = (x_1, \ldots, x_n) \in [d]^n$, and any adversary M, if we execute $L \leftarrow \mathrm{Manip}_{m,n}(\mathit{HH}_{n,k,\varepsilon}, \vec{x}, M)$ with parameter $k \leftarrow 3n^2/\beta$, then with probability $\ge 1 - \beta$, L contains all j such that*

$$\mathit{freq}(j, \vec{x}) > c \cdot \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \left(\sqrt{\frac{\log d}{n} \log \frac{n \log d}{\beta}} + \frac{m \log d}{n} \right)$$

In (Bassily et al., 2017), the authors show that it in fact suffices to take $k = O(1/\alpha) = \tilde{O}(\sqrt{n})$, which means a smaller list and faster running time is achievable. We choose a larger k to simplify some arguments.

There are three undesirable events that can occur when the game is played. In the claims below, we state them formally and bound the probability of each event by $\beta/3$. We first consider the event that the frequency of any $j \in \vec{x}$ is significantly different from the frequency of $j \in \vec{x}^{(g)}$:

Claim C.2. Fix any $\vec{x} \in [d]^n$. There is a constant c such that, when π is a uniformly random partition of $[n]$ into groups $\pi(1), \ldots, \pi(\log_2 d)$ each of size $n/\log_2 d$,

$$\mathbb{P}\left[\forall j \in \vec{x}\ \forall g \in [\log_2 d]\ |\mathit{freq}(j, \vec{x}) - \mathit{freq}(j, \vec{x}^{(g)})| > c \cdot \sqrt{\frac{\log_2 d}{n} \ln \frac{n \cdot \log_2 d}{\beta}} \right] \le \beta/3$$

This is proven via a Hoeffding bound and a union bound. Next we argue that there are likely no collisions:

Claim C.3. If $k > 3n^2/\beta$, then for any $\vec{x} \in [d]^n$ and a uniformly chosen $h : [d] \to [k]$,

$$\mathbb{P}\big[\exists x \ne x' \in \vec{x}\ h(x) = h(x')\big] \le \beta/3$$

Proof. The argument is brief:

$$\begin{aligned}\mathbb{P}\big[\exists x \neq x' \in \vec{x}\ h(x) = h(x')\big] &\leq n \cdot \mathbb{P}[\exists x \neq x_1\ h(x) = h(x_1)] \\ &\leq n^2 \cdot \mathbb{P}[h(x_2) = h(x_1)] \\ &= \frac{n^2}{k} < \beta/3\end{aligned}$$

□

A core part of the protocol is, for each group g, the execution of $\texttt{EST1}_{n',2k,\varepsilon}$ on one-hot encodings $(x'_i)_{i\in\pi(g)}$. Theorem B.9 implies the following:

Claim C.4. Fix any $m < n'$, $\vec{x}\,' \in (B_1^{2k})^{n'}$, any adversary M against $\texttt{EST1}_{n',2k,\varepsilon}$, and any $\beta \in (0,1)$. There exists a constant c such that

$$\mathbb{P}\left[\left\|\text{Manip}_{m,n'}(\texttt{EST1}_{n',2k,\varepsilon}, \vec{x}\,', M) - \frac{1}{n'}\sum_{i=1}^{n'} x'_i\right\|_\infty > c \cdot \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \left(\sqrt{\frac{1}{n'}\ln\frac{k}{\beta}} + \frac{m}{n'}\right)\right] \leq \beta/3$$

We are now ready to prove Theorem C.1

Proof of Theorem C.1. Let c_0, c_1 be the constants from Claims C.2 and C.4, respectively. We will prove that with probability $\geq 1-\beta$, for each $g \in [\log_2 d]$ and for each j such that

$$\mathit{freq}(j,\vec{x}) > c_0 \cdot \sqrt{\frac{\log_2 d}{n}\ln\frac{n\cdot\log_2 d}{\beta}} + 2c_1 \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \left(\sqrt{\frac{1}{n'}\ln\frac{k\log_2 d}{\beta}} + \frac{m}{n'}\right) \tag{C.1}$$

the protocol will reconstruct the g-th bit of j.

Each user constructs $x'_i \leftarrow \texttt{OneHotHash}_{h,k}(g(i), x_i)$ when running $R^{\texttt{HH}}$ on their data. A union bound over Claims C.2 and C.3 implies the following two inequalities hold for all groups g (with probability $1 - 2\beta/3$):

$$\frac{1}{n'}\sum_{i\in\pi(g)} x'_{i,2\cdot h(j)-bit(g,j)} \geq \mathit{freq}(j,\vec{x}) - c_0 \cdot \sqrt{\frac{\log_2 d}{n}\ln\frac{n\cdot\log_2 d}{\beta}} \tag{C.2}$$

$$\frac{1}{n'}\sum_{i\in\pi(g)} x'_{i,2\cdot h(j)+bit(g,j)-1} = 0 \tag{C.3}$$

From Claim C.4, the following two inequalities hold for all g (with probability $1-\beta/3$):

$$|z^{(g)}_{2\cdot h(j)-bit(g,j)} - \frac{1}{n'}\sum_{i\in\pi(g)} x'_{i,2\cdot h(j)-bit(g,j)}| \leq c_1 \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \left(\sqrt{\frac{1}{n'}\ln\frac{k\log_2 d}{\beta}} + \frac{m}{n'}\right) \tag{C.4}$$

$$|z^{(g)}_{2\cdot h(j)+bit(g,j)-1} - \frac{1}{n'}\sum_{i\in\pi(g)} x'_{i,2\cdot h(j)+bit(g,j)-1}| \leq c_1 \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1} \cdot \left(\sqrt{\frac{1}{n'}\ln\frac{k\log_2 d}{\beta}} + \frac{m}{n'}\right) \tag{C.5}$$

By a union bound, the following holds with probability $\geq 1-\beta$:

$$
\begin{aligned}
& z^{(g)}_{2\cdot h(j)-bit(g,j)} \\
& \geq \frac{1}{n'}\sum_{i\in\pi(g)} x'_{i,2\cdot h(j)-bit(g,j)} - c_1\cdot\frac{e^\varepsilon+1}{e^\varepsilon-1}\cdot\left(\sqrt{\frac{1}{n'}\ln\frac{k\log_2 d}{\beta}}+\frac{m}{n'}\right) && \text{(From (C.4))} \\
& \geq \mathit{freq}(j,\vec{x}) - c_0\cdot\sqrt{\frac{\log_2 d}{n}\ln\frac{n\cdot\log_2 d}{\beta}} - c_1\cdot\frac{e^\varepsilon+1}{e^\varepsilon-1}\cdot\left(\sqrt{\frac{1}{n'}\ln\frac{k\log_2 d}{\beta}}+\frac{m}{n'}\right) && \text{(From (C.2))} \\
& \geq c_1\cdot\frac{e^\varepsilon+1}{e^\varepsilon-1}\cdot\left(\sqrt{\frac{1}{n'}\ln\frac{k\log_2 d}{\beta}}+\frac{m}{n'}\right) && \text{(From (C.1))} \\
& \geq z^{(g)}_{2\cdot h(j)+bit(g,j)-1} + \frac{1}{n'}\sum_{i\in\pi(g)} x'_{i,2\cdot h(j)+bit(g,j)-1} && \text{(From (C.5))} \\
& = z^{(g)}_{2\cdot h(j)+bit(g,j)-1} && \text{(From (C.3))}
\end{aligned}
$$

By construction, this means we will assign $bit^{(g)}_{h(j)} \leftarrow bit(g,j)$. □