The Role of Interactivity in Local Differential Privacy

Matthew Joseph*, Jieming Mao†, Seth Neel‡, and Aaron Roth*
*Department of Computer and Information Science, University of Pennsylvania. majos, aaroth@cis.upenn.edu.
†Google Research New York. This done while at the University of Pennsylvania Warren Center. maojm@google.com.
‡Wharton School, Department of Statistics, University of Pennsylvania. sethneel@wharton.upenn.edu

***Abstract*—We study the power of interactivity in local differential privacy. First, we focus on the difference between fully interactive and sequentially interactive protocols. Sequentially interactive protocols may query users adaptively in sequence, but they cannot return to previously queried users. The vast majority of existing lower bounds for local differential privacy apply only to sequentially interactive protocols, and before this paper it was not known whether fully interactive protocols were more powerful.**

We resolve this question. First, we classify locally private protocols by their compositionality, the multiplicative factor by which the sum of a protocol's single-round privacy parameters exceeds its overall privacy guarantee. We then show how to efficiently transform any fully interactive compositional protocol into an equivalent sequentially interactive protocol with a blowup in sample complexity linear in this compositionality. Next, we show that our reduction is tight by exhibiting a family of problems such that any sequentially interactive protocol requires this blowup in sample complexity over a fully interactive compositional protocol.

We then turn our attention to hypothesis testing problems. We show that for a large class of compound hypothesis testing problems — which include all simple hypothesis testing problems as a special case — a simple noninteractive test is optimal among the class of all (possibly fully interactive) tests.

***Index Terms*—differential privacy; local differential privacy; interaction**

I. Introduction

In the last several years, differential privacy in the *local* model has seen wide adoption in industry, including at Google [6, 19], Apple [3], and Microsoft [14]. The choice of adopting the *local* model of differential privacy — in which privacy protections are added at each individual's device, before data aggregation — instead of the more powerful *central* model of differential privacy — in which a trusted intermediary is allowed to first aggregate data before adding privacy protections — is driven by practical concerns. Local differential privacy frees the data analyst from many of the responsibilities that come with the stewardship of private data, including liability for security breaches, and the legal responsibility to respond to subpoenas for private data, amongst others. However, the local model of differential privacy comes with its own practical difficulties. The most well known of these is the need to have access to a larger number of users than would be necessary in the central model. Another serious obstacle — the one we study in this paper — is the need for *interactivity*.

There are two reasons why interactive protocols — which query users adaptively, as a function of the answers to previous queries — pose practical difficulties. The first is that communication with user devices is slow: the communication in noninteractive protocols can be fully parallelized, but for interactive protocols, the number of rounds of interactivity becomes a running-time bottleneck. The second is that user devices can go offline or otherwise become unreachable — and so it may not be possible to return to a previously queried user and pose a new query. The first difficulty motivates the study of noninteractive protocols. The second difficulty motives the study of *sequentially interactive* protocols [16] — which may pose adaptively chosen queries — but must not pose more than one query to any user (and so in particular never need to return to a previously queried user).

It has been known since [26] that there can be an exponential gap in the sample complexity between noninteractive and interactive protocols in the local model of differential privacy, and that this gap can manifest itself even in natural problems like convex optimization [30, 31]. However, it was not known whether the full power of the local model could be realized with only *sequentially interactive* protocols. Almost all known lower bound techniques applied only to either noninteractive or sequentially interactive protocols, but there were no

2575-8454/19/$31.00 ©2019 IEEE
DOI 10.1109/FOCS.2019.00015

IEEE computer society

known fully interactive protocols that could circumvent lower bounds for sequential interactivity.

A. Our Results

We present two kinds of results, relating to the power of sequentially adaptive protocols and non-adaptive protocols respectively. Throughout, we consider protocols operating on datasets that are drawn i.i.d. from some unknown distribution $\mathcal{D}$, and focus on the *sample complexity* of these protocols: how many users (each corresponding to a sample from $\mathcal{D}$) are needed in order to solve some problem, defined in terms of $\mathcal{D}$.

a) Sequential Interactivity: We classify locally private protocols in terms of their *compositionality*. Informally, a protocol is k-compositional if the privacy costs $\{\varepsilon_j^i\}_{j=1}^r$ of the local randomizers executed by any user i over the course of the protocol sum to at most $k\varepsilon$, where ε is the overall privacy cost of the protocol: $\sum_j \varepsilon_j^i \leq k\varepsilon$. When $k = 1$, we say that the protocol is compositional. Compositional protocols capture most of the algorithms studied in the published literature, and in particular, any protocol whose privacy guarantee is proven using the composition theorem for ε-differential privacy[1].

1) **Upper Bounds**: For any (potentially fully interactive) compositional protocol M, we give a generic and efficient reduction that compiles it into a sequentially interactive protocol M', with only a constant factor blow-up in privacy guarantees and sample complexity, while preserving (exactly) the distribution on transcripts generated. This in particular implies that up to constant factors, sequentially adaptive compositional protocols are as powerful as fully adaptive compositional protocols. More generally, our reduction compiles an arbitrary k-compositional protocol M into a sequentially interactive protocol M' with the same transcript distribution, and a blowup in sample complexity of $O(k)$.
2) **Lower Bounds**: We show that our upper bound is tight by proving a separation between the power of sequentially and fully interactive protocols in the local model. In particular, we define a family of problems (Multi-Party Pointer Jumping) such that for any k, there is a fully interactive k-compositional protocol which can solve the problem given sample complexity $n = n(k)$, but such that no sequentially interactive protocol with the same privacy guarantees can solve the problem with sample complexity $\tilde{o}(k \cdot n)$. Thus, the sample complexity blowup of our reduction cannot be improved in general.

[1]Not every protocol is 1-compositional: exceptions include RAPPOR [19] and the evolving data protocol of Joseph et al. [24].

b) Noninteractivity: We then turn our attention to the power of noninteractive protocols. We consider a large class of compound hypothesis testing problems — those such that both the null hypothesis H_0 and the alternative hypothesis H_1 are closed under mixtures. For every problem in this class, we show that the optimal locally private hypothesis test is noninteractive. We do this by demonstrating the existence of a simple hypothesis test for such problems. We then prove that this test's sample complexity is optimal even among the set of all fully interactive tests by extending information theoretical lower bound techniques developed by Braverman et al. [8] and first applied to local privacy by Joseph et al. [23] and Duchi and Rogers [15] to the fully interactive setting.

B. Related Work

The local model of differential privacy was introduced by Dwork et al. [18] and further formalized by Kasiviswanathan et al. [26], who also gave the first separation between noninteractive locally private protocols and interactive locally private protocols. They did so by constructing a problem, Masked Parity, that requires exponentially larger sample complexity without interaction than with interaction. Daniely and Feldman [13] later expanded this result to a larger class of problems. Smith et al. [30] proved a similar separation (which applies generally to local oracle-based protocols) between noninteractive and interactive locally private convex optimization protocols.

Recent work by Acharya et al. [1, 2] gives a qualitatively different separation between the private-coin and public-coin models of noninteractive local privacy. Informally, the public-coin model allows for an additional "half step" of interaction over the private-coin model in the form of coordinated local randomizer choices across users. In this paper, we use the public-coin model of noninteractivity.

Duchi et al. [16] introduced the notion of sequential interactivity for local privacy. They also provided the first general techniques for proving lower bounds for sequentially interactive locally private protocols by bounding the KL-divergence between the output distributions of ε-locally private protocols with different input distributions as a function of ε and the total variation distance between these input distributions. Bassily and Smith [5] and Bun et al. [7] later generalized this result to (ε, δ)-locally private protocols, and Duchi et al. [17] obtained an

analogue of Assouad's method for proving lower bounds for sequentially interactive locally private protocols.

More recently, Duchi and Rogers [15] showed how to combine the above analogue of Assouad's method with techniques from information complexity [8, 21] to prove lower bounds for estimation problems that apply to a restricted class of *fully* interactive locally private protocols. A corollary of their lower bounds is that several known *noninteractive* algorithms are optimal minimax estimators within the class they consider. However, the class of protocols they study does not capture *non*-compositional locally private algorithms (details appear in the full version of this paper [25]). Our reduction implies that every (arbitrarily interactive) compositional locally private algorithm can be reduced to a sequentially interactive protocol with only constant blowup in sample complexity, and as a result all known lower bounds for sequentially interactive protocols also hold for arbitrary compositional protocols.

Canonne et al. [10] study simple hypothesis testing under the centralized model of differential privacy, and Theorem 1 of Duchi et al. [16] implies a tight lower bound for sequentially interactive locally private simple hypothesis testing. We extend this lower bound to the fully interactive setting and match it with a noninteractive upper bound for a more general class of compound testing problems that includes simple hypothesis testing as a special case.

II. Preliminaries

We begin with the definition of approximate differential privacy. Given data domain $\mathcal{X}$, two data sets $S, S' \in \mathcal{X}^n$ are *neighbors* (denoted $S \sim S'$) if they differ in at most one coordinate: i.e. if there exists an index i such that for all $j \neq i$, $S_j = S'_j$. A differentially private algorithm must have similar output distributions on all pairs of neighboring datasets.

Definition II.1 ([18])**.** *Let $\varepsilon, \delta \geq 0$. A randomized algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{O}$ is* (ε, δ)-differentially private *if for every pair of neighboring data sets $S \sim S' \in \mathcal{X}^n$, and every event $\Omega \subseteq \mathcal{O}$*

$$\mathbb{P}_{\mathcal{M}}\left[\mathcal{A}(S) \in \Omega\right] \leq \exp(\varepsilon)\mathbb{P}_{\mathcal{M}}\left[\mathcal{A}(S') \in \Omega\right] + \delta.$$

When $\delta = 0$, we say that $\mathcal{M}$ satisfies (pure) ε-differential privacy.

Differential privacy has two nice properties. First, it composes neatly: the composition of algorithms $\mathcal{M}_1, \ldots, \mathcal{M}_n$ that are respectively $(\varepsilon_1, \delta_1), \ldots, (\varepsilon_n, \delta_n)$-differentially private is $(\sum_i \varepsilon_i, \sum_i \delta_i)$-differentially private. For pure differential privacy, this is tight in general. Second, differential privacy is resilient to post-processing: given an (ε, δ)-differentially private $\mathcal{M}$ and any function f, $f \circ \mathcal{M}$ is still (ε, δ)-differentially private. For brevity, we often abbreviate "differential privacy" as "privacy".

As defined, the constraint of differential privacy is on the *output* of an algorithm $\mathcal{M}$, not on its internal workings. Hence, it implicitly assumes a trusted data curator, who has access to the entire raw dataset. This is sometimes referred to as differential privacy in the central model. In contrast, this paper focuses on the more restrictive *local* model [18] of differential privacy. In the local model, the private computation is an interaction between n users, each of whom hold exactly one dataset record, and is coordinated by a protocol $\mathcal{A}$. We assume throughout this paper that each user's datum is drawn i.i.d. from some unknown distribution: $x_i \sim_{iid} \mathcal{D}$[2]. Informally, at each round t of the interaction, a protocol $\mathcal{A}$ observes the transcript of interactions so far, selects a user, and assigns the user a randomizer. The user then applies the randomizer to their datum, using fresh randomness for each application, and publishes the output. In turn, the protocol observes the updated transcript, selects a new user-randomizer pair, and the process continues. We define these terms precisely below.

Definition II.2. *An* (ε, δ)-randomizer $R\colon X \to Y$ *is an* (ε, δ)*-differentially private function taking a single data point as input.*

A simple, canonical, and useful randomizer is *randomized response* [18, 32].

Example II.1 (Randomized Response)**.** *Given data universe $\mathcal{X} = [k]$ and datum $x_i \in \mathcal{X}$, ε-randomizer $\mathsf{RR}(x_i, \varepsilon)$ outputs x_i with probability $\frac{e^\varepsilon}{e^\varepsilon + k - 1}$ and otherwise outputs a uniformly random element of $\mathcal{X} - \{x_i\}$.*

Next, we formally define transcripts and protocols.

Definition II.3. *A* transcript π *is a vector consisting of 5-tuples $(i^t, R_t, \varepsilon_t, \delta_t, y_t)$ — encoding the user chosen, randomizer assigned, randomizer privacy parameters, and randomized output produced — for each round t. $\pi_{<t}$ denotes the transcript prefix before round t. Letting S_π denote the collection of all transcripts and S_R the collection of all randomizers, a* protocol *is a function $\mathcal{A}\colon S_\pi \to ([n] \times S_R \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}) \cup \{\perp\}$ mapping transcripts to users, randomizers, and randomizer pri-*

[2]Roughly speaking, this corresponds to a setting in which users are "symmetric" and in which nothing differentiates them a priori. All of our results generalize to the setting in which there are different "types" of users, known to the protocol up front.

vacy parameters ($\perp$ is a special character indicating a protocol halt).

The transcript that results from running a locally private computation will often be post-processed to compute some useful function of the data. However, the privacy guarantee must hold even if the entire transcript is observed. Hence, in this paper we abstract away the task that the computation is intended to solve, and view the output of a locally private computation as simply the transcript it generates.

To clarify the role of interaction in these private computations – especially when analyzing reductions between computations with different kinds of interactivity – it is often useful to speak separately of protocols and *experiments*. While the protocol $\mathcal{A}$ is a function mapping transcripts to users and randomizers, the experiment is the interactive process that maps a protocol and collection of users drawn from a distribution $\mathcal{D}$ to a finished transcript. In the simplest case, FollowExpt (Algorithm 1), the experiment exactly follows the outputs of its protocol.

Algorithm 1

procedure $\mathsf{FollowExpt}(\mathcal{A}, \mathcal{D}, n)$
  Draw n users $\{x_i\} \sim \mathcal{D}^n$
  Initialize transcript $\pi_0 \leftarrow \emptyset$
  for $t = 1, 2, \ldots$ **do**
    if $\mathcal{A}(\pi_{<t}) = \perp$ **then**
      Output transcript $\pi_{<t}$
    else
      $(i^t, R_t, \varepsilon_t, \delta_t) \leftarrow \mathcal{A}(\pi_{<t})$
      User i^t publishes $y_t \sim R_t(x_{i^t}, \varepsilon_t, \delta_t)$
    end if
  end for
end procedure

However, experiments may in general heed, modify, or ignore the outputs of their input protocol. We delineate the privacy characteristics of experiment-protocol pairs and protocols in isolation below. Here and throughout, the dataset is not viewed as an input to an experiment, but is drawn from $\mathcal{D}$ by the experiment-protocol pair. Drawing a fresh user $\sim \mathcal{D}$ corresponds to adding an additional data point, and so the sample complexity of an experiment-protocol pair is the number of draws from $\mathcal{D}$ over the run of the algorithm. For the simple algorithm $\mathsf{FollowExpt}(\mathcal{A})$ defined above, the sample complexity is always n. Finally we remark that although the distribution $\mathcal{D}$ and the sample complexity n are inputs to the experiment, for brevity we typically omit them and focus on the protocol $\mathcal{A}$; e.g. writing $\mathsf{Expt}(\mathcal{A})$ rather than $\mathsf{Expt}(\mathcal{A}, \mathcal{D}, n)$.

Definition II.4. *Experiment-protocol pair* $\mathsf{Expt}(\mathcal{A})$ *satisfies* (ε, δ)-local differential privacy *(LDP) if it is* (ε, δ)*-differentially private in its transcript outputs. A protocol* $\mathcal{A}$ *satisfies* (ε, δ)*-local differential privacy (LDP) if experiment-protocol pair* $\mathsf{FollowExpt}(\mathcal{A})$ *is* (ε, δ)*-locally differentially private.*

Experiment-protocol pairs can be, by increasing order of generality, *noninteractive*, *sequentially interactive*, and *fully interactive*.

Definition II.5. *An experiment-protocol pair* $\mathsf{Expt}(\mathcal{A})$ *is* noninteractive *if, at each round* t*, as random variables,* $(i^t, R_t, \varepsilon_t, \delta_t) \perp\!\!\!\perp \Pi_{<t} \mid t$.

In other words, noninteractivity forces nonadaptivity, and all user-randomizer assignments are made before the experiment begins. In contrast, in sequentially interactive experiment-protocol pairs, users may be queried adaptively, but only once.

Definition II.6. *An experiment-protocol pair* $\mathsf{Expt}(\mathcal{A})$ *is* sequentially interactive *if, at each round* t*,* $i^t \neq i^{t-1}, \ldots, i^1$.

Finally, in in fully interactive experiments, the experiment-protocol may make user-randomizer assignments adaptively, and each user may receive arbitrarily many randomizer assignments. Along the same lines, we say a protocol $\mathcal{A}$ is noninteractive (respectively sequentially and fully interactive) if $\mathsf{FollowExpt}(\mathcal{A})$ is a noninteractive (respectively sequentially and fully interactive) experiment-protocol pair. This experiment-protocol formalism will be useful in constructing the full-to-sequential reduction in Section III; elsewhere, we typically elide the distinction and simply reason about $\mathsf{FollowExpt}(\mathcal{A})$ as "protocol $\mathcal{A}$". For any locally private protocol, we refer to the number of users n that it queries as its *sample complexity*. For fully interactive protocols, the total number of rounds — which we denote by T — may greatly exceed n. In contrast, for both noninteractive and sequentially interactive protocols, the number of rounds $T = n$.

At each round t of a fully interactive ε-locally private protocol, we know that $\varepsilon_t \leq \varepsilon$. For many protocols, we can say more about how the ε_t parameters relate to ε:

Definition II.7. *Consider an* ε*-locally private protocol* $\mathcal{A}$*. Let* $\{\varepsilon_t\}_{t=1}^T$ *denote the minimal privacy parameters of the local randomizers* R_t *selected at round* t *considered as random variables. We say the protocol* $\mathcal{A}$ *is* k-

compositionally private *if for all* $i \in [n]$*, with probability* 1 *over the randomness of the transcript,*

$$\sum_{t:\ i_t = i} \varepsilon_t \leq k\varepsilon.$$

If $k = 1$*, a protocol is simply* compositional private*.*

Remark. *In fact, all of our results hold without modification even under the weaker condition of* average k*-compositionality. For a protocol* $\mathcal{A}$ *with sample complexity* n*,* $\mathcal{A}$ *is* k*-compositional on average if*

$$\sum_{t} \varepsilon_t \leq k\varepsilon n.$$

For brevity, we often shorthand "k*-compositionally private" as simply "*k*-compositional".*

Informally, a compositionally private protocol is one in which the privacy parameters for each user "just add up." Almost every locally private protocol studied in the literature (and in particular, every protocol whose privacy analysis follows from the composition theorem for pure differential privacy) is compositionally private[3]. They are so ubiquitous that it is tempting to guess that all $(\varepsilon, 0)$-locally private protocols are compositional. However, this is false: for every k and ε, there are ε-locally private protocols that fail to be k-compositionally private. The following example shows that by taking advantage of special structure in the data domain and choice of randomizers it is possible to achieve $(\varepsilon, 0)$-local privacy, even as the sum of the round-by-round privacy parameters greatly exceeds ε.

Example II.2 (Informal)**.** *Let the data universe* $\mathcal{X}$ *consist of the canonical basis vectors* $e_1, \ldots, e_d \in \{0,1\}^d$*, and let each* $x_1, \ldots, x_n$ *be an arbitrary element of* $\mathcal{X}$*. Consider the* d *round protocol where, for each round* $j \in [d]$*, every user* i *with* $x_i = e_j$ *outputs a sample from* $\mathsf{RR}(1, \varepsilon)$*, and the remaining users output a sample from* $\mathsf{Ber}(0.5)$*. As* $\mathsf{RR}(\cdot, \varepsilon)$ *is an* ε*-local randomizer which each user employs only once, and remaining outputs are data-independent, this protocol is* ε*-locally private. But the protocol fails to be* k*-compositionally private for* $k < d/2$*.*

The preceding example demonstrates that the careful choice of local randomizers based on the data universe structure can strongly violate compositional privacy. Seen another way, when multiple queries are asked of the same user, there are situations in which the correlation in privatized responses induced by being run on the same data element can lead to arbitrarily sub-compositional privacy costs. The main result of our paper is that the additional power of a fully interactive protocol, on top of sequential interactivity, is characterized by its compositionality.

[3]This simple compositionality applies even if $\{\varepsilon_t\}_{t=1}^T$ are chosen adaptively in each round (see Theorem 3.6 in Rogers et al. [29]).

III. From Full to Sequential Interactivity

We show that any $(\varepsilon, 0)$-locally private compositional protocol is "equivalent" to a sequentially interactive protocol with sample complexity that is larger by only a small constant factor. By equivalent, we mean that for any $(\varepsilon, 0)$-locally private compositional protocol, we can exhibit a sequentially interactive $(3\varepsilon, 0)$-locally differentially private protocol with only a constant factor larger sample complexity that induces exactly the same distribution on transcripts. Thus for any task for which the original protocol was useful, the sequentially interactive protocol is just as useful[4].

More generally, we give a generic reduction under which any $(\varepsilon, 0)$-private k-compositional protocol can be compiled into a sequentially interactive protocol with an $e^{\varepsilon}k$-factor increase in sample complexity.

Our proof is constructive; given an arbitrary k-compositional $(\varepsilon, 0)$-locally differentially private protocol we show how to simulate it using a sequentially interactive protocol that induces the same joint distribution on transcripts. The "simulation" is driven by three main ideas:

1) **Bayesian Resampling**: The dataset used in a locally differentially private protocol is static once the protocol begins. However, we consider the following thought experiment: each user's datum is *resampled* from the posterior distribution on their datum, conditioned on the transcript thus far, before every round in which they are given a local randomizer. We observe that the mechanism from this thought experiment induces exactly the same joint distribution on datasets and transcripts

[4]Formally, for any loss function defined over a data distribution $\mathcal{D}$ and a transcript Π, when data points x_i are drawn i.i.d. from $\mathcal{D}$, the two protocols induce exactly the same distribution over transcripts, and hence the same distribution over losses. Once one restricts attention to locally private protocols with privacy parameter $\varepsilon \leq 1$ that take as input points drawn i.i.d. from a distribution $\mathcal{D}$, it is without loss of generality to measure the success or failure of a protocol with respect to the underlying distribution $\mathcal{D}$, rather than with respect to the sample. This is because such protocols are $\approx \varepsilon/\sqrt{n}$ differentially private when viewed in the central model of differential privacy (in which the input may be permuted before used in the protocol) [4, 20], and hence the distribution on transcripts would be almost unchanged even if the entire dataset was *resampled* i.i.d. from $\mathcal{D}$. [12, 27]. Thus, for such protocols, the transcript distribution is governed by the data distribution $\mathcal{D}$, but not (significantly) by the sample.

upon completion of the mechanism. Thus, for the remainder of the argument, we can seek to simulate this "Bayesian Resampling" version of the mechanism.

2) **Private Rejection Sampling**: Because of the local differential privacy guarantee, at any step of the algorithm, the posterior on a user's datum conditioned on the private transcript generated so far must be close to their prior. Thus, it is possible to sample from this posterior distribution by first sampling from the prior, and then applying a rejection sampling step that is both a) likely to succeed, and b) differentially private. Sampling from the prior simply corresponds to querying a new user. At first glance, applying rejection sampling as needed seems to require information that the users will not have available, because they do not know the underlying data distribution $\mathcal{D}$. But an application of Bayes rule, together with a data independent rescaling can be used to rewrite the required rejection probability using only quantities that each user can compute from her own data point and the transcript. A similar use of rejection sampling appears in the simulation of locally private algorithms by statistical query algorithms given by Kasiviswanathan et al. [26].
3) **Data Independent Decomposition of Local Randomizers**: The two ideas above suffice to transform a fully interactive mechanism into a sequentially interactive mechanism, with a blowup in sample complexity from n to T (because in the sequentially interactive protocol that results from rejection sampling, each user applies only one local randomizer instead of an average of T/n). However, we generalize a recent result of [4] to show that any ε_i-private local randomizer can be described as a mixture between a *data independent* distribution and an $(\varepsilon, 0)$-private local randomizer for any $\varepsilon > \varepsilon_i$, where the weight on the data independent distribution is roughly (for small constant ε) $1 - \varepsilon_i/\varepsilon$. Thus we can simulate each local randomizer while only needing to query a new user with probability $\varepsilon_i/\varepsilon$. As a result, for any compositional mechanism, 1 user in the sequential setting suffices (in expectation) to simulate the entire transcript of a single user in the fully interactive setting. More generally, if the mechanism is k-compositional, then k users are required in expectation to carry out the simulation. The realized sample complexity concentrates sharply around its expectation.

A. Step 1: A Bayesian Thought Experiment

The first step of our construction is to observe that for any locally private protocol $\mathcal{A}$, $\mathsf{BayesExpt}(\mathcal{A})$ induces exactly the same distribution over transcripts as $\mathsf{FollowExpt}(\mathcal{A})$. The difference is that in $\mathsf{BayesExpt}(\mathcal{A})$, between each interaction with a given user i, their datum x_i is *resampled* from the posterior distribution on user i's data conditioned on the portion of the transcript generated thus far. We prove in Lemma III.1 that the two experiments produce exactly the same transcript distribution. Once we establish this, our goal will be to simulate the transcript distribution induced by $\mathsf{BayesExpt}(\mathcal{A})$.

Algorithm 2

procedure $\mathsf{BayesExpt}(\mathcal{A}, \mathcal{D}, n)$
  Initialize transcript $\pi_0 = \emptyset$
  for $t = 1, 2, \ldots$ **do**
    if $\mathcal{A}(\pi_{<t}) = \perp$ **then**
      Output transcript $\pi_{<t}$
    else
      $(i^t, R_t, \varepsilon_t, \delta_t) \leftarrow \mathcal{A}(\pi_{<t})$
      Redraw $x_{i^t} \sim Q_{i,t}$
      User i^t publishes $y_t \sim R_t(x_{i^t})$
    end if
  end for
end procedure

Note that when i^t is selected for the first time $Q_{i,t} = \mathcal{D}$, and so the sample complexity (e.g. number of draws from $\mathcal{D}$) of $\mathsf{BayesExpt}(\mathcal{A})$ is bounded by n. For Lemma III.1 and in general, we defer all proofs and detailed pseudocode to the full version of this paper [25].

Lemma III.1. *For any protocol $\mathcal{A}$, Let Π^f be the transcript random variable that is output by $\mathsf{FollowExpt}(\mathcal{A})$ and let Π^b be the transcript output by $\mathsf{BayesExpt}(\mathcal{A})$. Then*

$$\Pi^f \stackrel{d}{=} \Pi^b$$

where $\stackrel{d}{=}$ denotes equality of distributions.

B. Step 2: Sequential Simulation of Algorithm 2 via Rejection Sampling

We now show how to replace step 8 in Algorithm 2 by selecting a new datapoint (drawn from $\mathcal{D}$) at every round and using rejection sampling to simulate a draw from $Q_{i,t}$. The result is a sequentially interactive mechanism that preserves the transcript distribution of Algorithm 2 (and, by Lemma III.1, of Algorithm 1), albeit one with a potentially very large increase in sample complexity

(from n to T). The rejection sampling step increases the privacy cost of the protocol by at most a factor of 2.

We first review why it is non-obvious that rejection sampling can be performed in this setting. We want to sample from the target distribution $Q_{i,t}$, the posterior $x_i^t|\pi_{<t}$, using samples from the proposal distribution $\mathcal{D}$. Let p_π denote the density function of $Q_{i,t}$ and let p denote the density function of $\mathcal{D}$. In rejection sampling, we would typically sample $u \sim \mathcal{D}$, and with probability $\propto \frac{p_\pi(u)}{p(u)}$ we would accept u as a sample drawn from $Q_{i,t}$, or else redraw another u and continue.

This is not immediately possible in our setting, since the individuals (who must perform the rejection sampling computation) do not know the prior density p and hence do not know the posterior p_π. As a result, they cannot compute either the numerator or denominator of the expression for the acceptance probability. We solve this problem by using the fact that we are simulating a posterior with a prior distribution, and formulate the rejection sampling probability ratio as a quantity depending only on a user's private data point and the transcript. Users may then compute this quantity themselves.

To define our transformed rejection sampler we set up some new notation: given a user i and round t, let $\pi_{<t,i}$ denote the subset of the realized transcript up to time t that corresponds to user i's data, i.e. $\pi_{<t,i} = \{(i^{t'}, R_{t'}, \varepsilon_{t'}, \delta_{t'}, y_{t'}) : t' < t, i^{t'} = i\}$. Let $\mathbb{P}_{x_i}[\pi_{<t,i}]$ denote the conditional probability of the messages corresponding to user i given the choices of privacy parameters and randomizers up to time t:

$$\mathbb{P}[\pi_{<t,i}] = \prod_{t' \,:\, i^{t'}=i} \mathbb{P}_{R_{t'}}[R_{t'}(x_i, \varepsilon_{t'}, \delta_{t'}) = y_{t'}].$$

Using this notation, we define our rejection sampling procedure RejSamp in Algorithm 3.

Algorithm 3 Rejection Sampling

procedure RejSamp($i, \pi_{<t}, \varepsilon, \varepsilon_t, R_t(\cdot), \mathcal{D}$)
  Initialize indicator accept $\leftarrow 0$
  while accept $= 0$ **do**
    Draw a new user $x \sim \mathcal{D}$
    User x computes $p_x \leftarrow \frac{\mathbb{P}_x[\pi_{<t,i}]}{\max_{x^*} \mathbb{P}_{x^*}[\pi_{<t,i}]}$
    User x publishes accept $\sim$ Ber$(p_x/2)$
    if accept $= 1$ **then**
      User x outputs $Y_t' \sim R_t(x, \varepsilon_t)$
    end if
  end while
end procedure

We now show that RejSamp is private and does not need to sample many users.

Lemma III.2. *Let $Y_t \overset{d}{=} R_t(x')$, where $x' \sim Q_{i,t}$ and let Y_t' be defined by the rejection sampling algorithm* RejSamp *above. Let the sample complexity N be the total number of new users x drawn in step 4 of* RejSamp*. Then* RejSamp *is $(\varepsilon+\varepsilon_t, 0)$-locally private, $Y_t \overset{d}{=} Y_t'$, and $\mathbb{E}[N] \le 2e^\varepsilon$.*

C. Step 3: Data Independent Decomposition of Local Randomizers

The preceding sections enable us to simulate a fully interactive k-compositional $(\varepsilon, 0)$-locally private protocol with a sequentially interactive $(2\varepsilon, 0)$-locally private protocol. However, our solution so far may require sampling a new user for each query in the original protocol. Since a fully interactive protocol's query complexity may greatly exceed its sample complexity, this is undesirable. To address this problem, we *decompose* each local randomizer in a way that substantially reduces the number of queries that actually require samples.

Let $R : \mathcal{X} \to \mathcal{Y}$ be an ε' local randomizer, fix an arbitrary element $x_0 \in \mathcal{X}$, and let x be a private input to R. Then Lemma 5.2 in Balle et al. [4] shows that we can write $R(x)$ as a mixture $\gamma w + (1-\gamma)d_x$, where w is a data-independent distribution, d_x is a data-dependent distribution, and $\gamma \ge e^{-\varepsilon'}$. This suggests that decomposition — by answering a proportion of queries from data-independent distributions — can reduce the sample complexity of our solution. Unfortunately, the data dependent distribution need not be differentially private (in fact, it often corresponds to a point mass on the private data point), so the privacy of the overall mechanism crucially relies on not releasing *which* of the two mixture distributions the output was sampled from.

We first generalize this result, showing that for any $\varepsilon \ge \varepsilon'$, we can write $R(x)$ as $(1-\gamma)w + \gamma\tilde{R}(x)$ where $\tilde{R}$ is a 2ε-differentially private local randomizer, and $\gamma = \frac{e^{-\varepsilon'}-1}{e^{-\varepsilon}-1}$ (Lemma III.3). The upshot of this generalization is that even if we make public which part of the mixture distribution was used, the resulting privacy loss is still bounded by 2ε. Larger values of ε increase our chance of sampling from a data-independent distribution when simulating a local randomizer, while increasing the privacy cost incurred by a user in the event that we sample from the data-dependent mixture component. This tradeoff will be crucial for us in the proof of our main result.

Lemma III.3 (Data Independent Decomposition)**.** *Let $R : \mathcal{X} \to \mathcal{Y}$ be an ε'-differentially private local randomizer and let $\varepsilon \ge \varepsilon'$. Then there exists a mapping*

$\tilde{R}$ and fixed data-independent distribution μ such that $\tilde{R}(\cdot)$ is a $2\varepsilon-$differentially private local randomizer and

$$R(x) \stackrel{d}{=} \gamma\tilde{R}(x) + (1-\gamma)\mu,$$

where $\gamma = \frac{e^{-\varepsilon'}-1}{e^{-\varepsilon}-1}$.

D. Putting it All Together: The Complete Simulation

Finally, we combine rejection sampling and decomposition to give our complete reduction, Algorithm 4. We use rejection sampling to convert from a fully interactive mechanism to a sequentially interactive one and use our data-independent decomposition of local randomizers to reduce the sample complexity of the converted mechanism.

Algorithm 4 Reduction

1: **procedure** Reduction(Fully interactive $(\varepsilon, 0)-$LDP Protocol $\mathcal{A}, \mathcal{D}, n$)
2: Initialize $s_1, \ldots, s_n \leftarrow 0$.
3: **for** $t = 1 \ldots$ **do**
4: **if** $\mathcal{A}(\pi_{<t}) = \perp$ **then**
5: Output transcript $\pi_{<t}$
6: **else**
7: $(i^t, R_t, \varepsilon_t) \leftarrow \mathcal{A}(\pi_{<t})$
8: **if** $s_i^t = 1$ **then**
9: Let $\gamma \leftarrow \frac{e^{-\varepsilon_t}-1}{e^{-\varepsilon}-1}$
10: Let $R_t = \gamma\tilde{R}_t + (1-\gamma)R_t(x_0)$
11: Draw $\rho \sim \mathsf{Unif}(0,1)$
12: **if** $\rho \leq \gamma$ **then**
13: Draw $Y_t \sim \mathsf{RejSamp}(i^t, \pi_{<t}, \varepsilon, 2\varepsilon, \tilde{R}(\cdot), \mathcal{D})$
14: **else**
15: Draw $Y_t \sim R_t(x_0, \varepsilon_t)$
16: **end if**
17: **else**
18: Draw $x_{i^t} \sim Q_{i,t} = \mathcal{D}$
19: Draw $Y_t \sim R_t(x_{i^t}, \varepsilon_t)$
20: Let $s_{i^t} \leftarrow 1$
21: **end if**
22: **end if**
23: **end for**
24: **end procedure**

We now show that Reduction has the desired interactivity, privacy, transcript, and sample complexity guarantees. We again denote by N the sample complexity of Reduction, i.e. the number of samples drawn from the prior $\mathcal{D}$ over the run of the algorithm, either in Step 15 (which is bounded by n), or over the runs of RejSamp in line 10. We observe that sampling from the prior $\mathcal{D}$ simply corresponds to using a new datapoint drawn from $\mathcal{D}$. Fixing a protocol $\mathcal{A}$, let Π^r denote the transcript random variable generated by $\mathsf{Reduction}(\mathcal{A})$, and let Π^b denote the transcript random variable generated by $\mathsf{BayesExpt}(\mathcal{A})$.

Theorem III.4. *Let $\mathcal{A}$ a fully-interactive k-compositional $(\varepsilon, 0)$-locally private protocol. Then*

1) $\mathsf{Reduction}(\mathcal{A})$ *is sequentially interactive,*
2) $\mathsf{Reduction}(\mathcal{A})$ *is $(3\varepsilon, 0)$-locally private,*
3) $\Pi^r \stackrel{d}{=} \Pi^b$,
4) $\mathbb{E}[N] \leq n(\frac{2e^{\varepsilon}\cdot\varepsilon}{1-e^{-\varepsilon}}k+1)$, *and with probability* $1-\delta$, $N = O(nk + \sqrt{nk\log\frac{1}{\delta}})$.

IV. Separating Full and Sequential Interactivity

We now prove that our reduction in Section III is tight in the sense that any generic reduction from a fully interactive protocol to a sequentially interactive protocol must have a sample complexity blowup of $\tilde{\Omega}(k)$ when applied to a k-compositional protocol. Specifically, we define a family of problems such that for every k, there is a fully interactive k-compositional protocol that can solve the problem with sample complexity $n = n(k)$, but such that *any* sequentially interactive protocol solving the problem must have sample complexity $\tilde{\Omega}(k \cdot n)$.

Informally, the family of problems (Multi-Party Pointer Jumping, or $\mathcal{MPJ}(d)$) we introduce is defined as follows. An *instance* of $\mathcal{MPJ}(d)$ is given by a complete tree of depth d. Every vertex of the tree is labelled by one of its children. By following the labels down the tree, starting at the root, an instance defines a unique root-to-leaf path. Given an instance of $\mathcal{MPJ}(d)$, the data distribution is defined as follows: to sample a new user, first select a level of the tree uniformly $\ell \in [d]$ at random, and provide that user with the vertex-labels corresponding to level ℓ (note that fixing an instance of the problem, every user corresponding to the same level of the tree has the same data). The problem we wish to solve privately is to identify the unique root to leaf path specified by the instance.

We first show that there is a fully interactive protocol which can solve this problem with sample complexity $n = \tilde{O}(d^2/\varepsilon^2)$. The protocol is k-compositional for $k = \Theta(d)$. Roughly speaking, the protocol works as follows: it identifies the path one vertex at a time, starting from the root, and proceeding to the leaf, in d rounds. In each round, given the most recently identified vertex v_i in level ℓ, it attempts to identify the child that vertex v_i is labelled with. It queries every user with the same

local randomizer, which asks them to use randomized response to identify the labelled child of v_i if their data corresponds to level ℓ, and to respond with a uniformly random child otherwise (recall that the level that a user's data corresponds to is itself private, and hence is not known to the protocol). Since there are roughly $\tilde{\Theta}(\sqrt{n}/\varepsilon^2)$ users with relevant data, out of n users total, it is possible to identitify the child in question subject to local differential privacy. Although every user applies an ε-local randomizer d times in sequence, because each user's data corresponds to only a single level in the tree, the protocol is still $(\varepsilon, 0)$-locally private. Note that this privacy analysis mirrors the "histogram" structure of the non-compositional protocol in Example II.2.

Informally, the reason that any sequentially interactive protocol must have sample complexity that is larger by a factor of d, is that even to identify the child of a single vertex in the local model, $\Omega(d^2/\varepsilon^2)$ datapoints are required (this is exactly what our randomized response protocol achieves). But a sequentially interactive protocol cannot re-use these datapoints across levels of the tree, and so must expend $\Omega(d^2/\varepsilon^2)$ samples for *each* of the d levels of the tree. This intuition is formalized in a delicate and technical induction on the depth of the tree, using information theoretic tools to bound the success probability of any protocol as a function of its sample complexity. The precise definition of $\mathcal{MPJ}(d)$ is somewhat more complicated, in which half of the weight on the underlying distribution is assigned to "level 0" dummy agents whose purpose is to break correlations between levels of the tree in the argument.

A. The Multi-Party Pointer Jumping Problem

We now formally define the *Multi-party Pointer Jumping* ($\mathcal{MPJ}$) problem.

Definition IV.1. *Given integer parameter $d > 1$, an instance of Multi-party Pointer Jumping $\mathcal{MPJ}(d)$ is defined by a vector $Z = Z_1 \circ \cdots \circ Z_d$, a concatenation of d vectors of increasing length. Letting $s = d^4$, for each $i \in [d]$ Z_i is a vector of s^{i-1} integers in $\{0, 1, \ldots, s-1\}$. For each Z_i, $Z_{i,j}$ is its j^{th} coordinate.*

Viewed as a tree, Z is a complete s-ary tree of depth d where each $Z_{i,j}$ marks a child of the j-th vertex at depth i. $P = P(Z)$ then denotes the vector of d integers representing the unique root to leaf path *down this tree through the children marked by Z. Formally, P is defined in a recursive way: $P_1 = Z_{1,1}$, ...,$P_i = Z_{i,P_1 \cdot s^{i-1} + P_2 \cdot s^{i-2} + \cdots + P_{i-1}+1}$,...,$P_d = Z_{d,P_1 \cdot s^{d-1} + P_2 \cdot s^{d-2} + \cdots + P_{d-1}+1}$.*

Finally, an instance $\mathcal{MPJ}(d)$ defines a data distribution $\mathcal{D}$. For each $x \sim \mathcal{D}$, with probability $1/2$, $x = (0, \emptyset)$ is a "dummy datapoint", and with the remaining probability $x = (\ell, Z_\ell)$ where ℓ is a level drawn uniformly at random from $[d]$. A protocol solves $\mathcal{MPJ}(d)$ if it recovers P using samples from $\mathcal{D}$.

Algorithm 5 A fully interactive $(\varepsilon, 0)$-locally private protocol for $\mathcal{MPJ}(d)$

Divide users into $u = \lceil \log(s)/\log(2) \rceil$ groups each of $m = 512 d^2 \log(d) \cdot \frac{(e^\varepsilon+1)^2}{(e^\varepsilon-1)^2}$ users.
Initialize $Q \leftarrow 0$
for $r = 1, 2, \ldots, d$ **do**
  $Q_r \leftarrow 0$
  for each group $g = 1, 2, \ldots, u$ **do**
    for each user $i = 1, 2, \ldots, m$ **do**
      $\ell_i \leftarrow$ level of user x_i
      if $\ell_i = r$ **then**
        $b_{i,r} \leftarrow g$-th bit of $Z_{r,Q+1}$
        Randomized response $y_i \sim \mathsf{RR}(b_{i,r}, \varepsilon)$
      else
        User i publishes $y_i \sim \mathsf{Ber}(0.5)$
      end if
    end for
    g-th bit of $Q_r \leftarrow$ majority bit of $\{y_i\}_{i=1}^m$
  end for
  $Q \leftarrow s \cdot Q + Q_r$
end for
Output $Q_1 \circ \cdots \circ Q_d$

B. An Upper Bound for Fully Interactive Mechanisms

Theorem IV.1. *There exists a fully interactive $(\varepsilon, 0)$-locally private protocol (Algorithm 5) with sample complexity $n = O(d^2 \log^2(d)(e^\varepsilon + 1)^2/(e^\varepsilon - 1)^2)$ that, on any instance Z of $\mathcal{MPJ}(d)$, correctly identifies $P(Z)$ with probability at least $1 - 1/d$.*

Note that Algorithm 5 is k-compositional only for $k \geq \Omega(d)$. The lower bound that we prove next (Theorem IV.2) shows that any sequentially interactive protocol for the same problem must have a larger sample complexity by a factor of $\tilde{\Omega}(d) = \tilde{\Omega}(k)$, showing that in general, the sample-complexity dependence that our reduction (Theorem III.4) has on k cannot be improved.

C. A Lower Bound for Sequentially Interactive Mechanisms

We prove our lower bound for sequentially interactive $(\varepsilon, 0)$-locally private protocols. As previous work [9,

11] has established that $(\varepsilon, 0)$- and (ε, δ)-local privacy are approximately equivalent for reasonable parameter ranges, our lower bound also holds for sequentially interactive (ε, δ)-locally private protocols. For an extended discussion of this equivalence, see Section V-A2.

Theorem IV.2. *Let $\mathcal{A}$ be a sequentially interactive $(\varepsilon, 0)$-locally private protocol that, for every instance Z of $\mathcal{MPJ}(d)$, correctly identifies $P(Z)$ with probability $\geq 2/3$. Then $\mathcal{A}$ must have sample complexity $n \geq d^3/(216(e^\varepsilon - 1)^2 \log(d))$.*

V. Hypothesis Testing

We now turn our attention to the role of interactivity in hypothesis testing. We first show that for the simple hypothesis testing problem, there exists a non-interactive $(\varepsilon, 0)$-LDP protocol that achieves optimal sample complexity. This result extends to the compound hypothesis testing case, when we make the additional assumption that the sets of distributions are convex and compact.

A. Simple Hypothesis Testing

Let P_0 and P_1 be two known distributions such that $\|P_0 - P_1\|_{TV} \geq \alpha$, and suppose one of P_0 and P_1 generates n i.i.d. samples $x_1, \ldots, x_n$. The goal in *simple hypothesis testing* is to determine whether the samples are generated by P_0 or P_1. The Neyman-Pearson lemma [28] establishes that the likelihood ratio test is optimal for this problem absent privacy, and recent work [10] extends this idea to give an optimal (up to constants) private simple hypothesis test in the centralized model of differential privacy. We recall a simple folklore non-interactive hypothesis test in the local model, and then prove that it is optimal even among the set of all fully interactive locally private tests.

1) (Folklore) Upper Bound: Consider the following simple variant $\mathcal{A}$ of the likelihood ratio test: each user i with input x_i outputs $\mathsf{RR}(\varepsilon) \arg\max_{j \in \{0,1\}} P_j(x_i)$. For $j \in \{0, 1\}$ let $\hat{N}_j$ denote the resulting count of responses and let $\hat{N}'_j = \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \left(\hat{N}_j - \frac{n}{e^\varepsilon - 1}\right)$ be the corresponding de-biased count. The analyst computes both quantities $\hat{N}'_j$ and outputs $P_{\arg\max_j \hat{N}'_j}$.

It is immediate that $\mathcal{A}$ is noninteractive and, since it relies on randomized response, satisfies $(\varepsilon, 0)$-local differential privacy. We can bound its sample complexity by simple concentration arguments.

Theorem V.1. *With probability at least $2/3$, $\mathcal{A}$ distinguishes between P_0 and P_1 given $n = \Omega\left(\frac{1}{\varepsilon^2 \alpha^2}\right)$ samples.*

Algorithm 6 Locally Private Simple Hypothesis Tester $\mathcal{A}$

procedure NonInteractive Protocol($\{x_i\}_{i=1}^n$)
  for $i = 1 \ldots n$ **do**
    $y_i \leftarrow \mathsf{RR}\left(\arg\max_{j \in \{0,1\}} P_j(x_i), \varepsilon\right)$
  end for
  for $j = 0, 1$ **do**
    Analyst computes $\hat{N}_j \leftarrow |\{y_i \mid y_i = j\}|$
    Analyst computes $\hat{N}'_j \leftarrow \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \left(\hat{N}_j - \frac{n}{e^\varepsilon - 1}\right)$
  end for
  Analyst outputs $P_{\arg\max_j \hat{N}'_j}$
end procedure

2) A Lower Bound for Arbitrarily Adaptive (ε, δ)-Locally Private Tests: We now show that the folklore ε-private non-interactive test is optimal amongst all (ε, δ)-private fully interactive tests. First, combining (slightly modified versions of) Theorem 6.1 from Bun et al. [9] and Theorem A.1 in Cheu et al. [11], we get the following result[5]

Lemma V.2. *Given $\varepsilon > 0$, $\delta < \min\left(\frac{\varepsilon\beta}{48n\ln(2n/\beta)}, \frac{\beta}{64n\ln(n/\beta)e^{7\varepsilon}}\right)$ and sequentially interactive (ε, δ)-locally private protocol $\mathcal{A}$, there exists a sequentially interactive $(10\varepsilon, 0)$-locally private protocol $\mathcal{A}'$ such that for any dataset U, $\|\mathcal{A}(U) - \mathcal{A}'(U)\|_{TV} \leq \beta$.*

Lemma V.2 enables us to apply existing lower bound tools for ε-locally private protocols to (sequentially interactive) (ε, δ)-locally private protocols. At a high level, our proof relies on controlling the Hellinger distance between transcript distributions induced by an (ε, δ)-locally private protocol when samples are generated by P_0 and P_1. We borrow a simulation technique used by Braverman et al. [8] for a similar (non-private) problem and find that we can control this Hellinger distance by bounding the KL divergence between a simpler, *noninteractive* pair of transcript distributions. We accomplish this last step using existing tools from Duchi et al. [16].

Theorem V.3. *Let $\|P_0 - P_1\|_{TV} = \alpha$ and let Π be an arbitrary (possibly fully interactive) (ε, δ)-locally private simple hypothesis testing protocol distinguishing be-*

[5] Bun et al. [9] and Cheu et al. [11] prove their results for noninteractive protocols. However, their constructions both rely on replacing a single (ε, δ)-local randomizer call for each user with an $(O(\varepsilon), 0)$-local randomizer call and proving that these randomizers induce similar output distributions. Since each user still makes a single randomizer call in sequential interactive protocols, essentially the same argument applies.

tween P_0 and P_1 with probability $\geq 2/3$ using n samples where $\varepsilon > 0$ and $\delta < \min\left(\frac{\varepsilon^3\alpha^2}{48n\ln(2n/\beta)}, \frac{\varepsilon^2\alpha^2}{64n\ln(n/\beta)e^{7\varepsilon}}\right)$. Then $n = \Omega\left(\frac{1}{\varepsilon^2\alpha^2}\right)$.

B. Compound Hypothesis Testing

We now extend the reasoning of Section V to *compound* hypothesis testing. Here P_0 and P_1 are replaced by (disjoint) collections of discrete hypotheses H_0 and H_1 such that

$$\inf_{(P,Q)\in H_0\times H_1} \|P - Q\|_{TV} \geq \alpha.$$

The goal is to determine whether samples are generated by a distribution in H_0 or one in H_1.

Theorem V.4. *Let H_0 and H_1 be convex and compact sets of distributions over ground set X such that $\inf_{(P,Q)\in H_0\times H_1} \|P - Q\|_{TV} \geq \alpha$. Then there exists noninteractive $(\varepsilon, 0)$-locally private protocol $\mathcal{A}$ that with probability at least $2/3$ distinguishes between H_0 and H_1 given $n = \Omega\left(\frac{1}{\varepsilon^2\alpha^2}\right)$ samples.*

Since Theorem V.3 still applies, this establishes that the above non-interactive protocol is also optimal.

References

[1] Jayadev Acharya, Clément Canonne, Cody Freitag, and Himanshu Tyagi. Test without trust: Optimal locally private distribution testing. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[2] Jayadev Acharya, Clément L Canonne, and Himanshu Tyagi. Inference under information constraints: Lower bounds from chi-square contraction. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 3–17, Phoenix, USA, 25–28 Jun 2019. PMLR. URL http://proceedings.mlr.press/v99/acharya19a.html.

[3] Differential Privacy Team Apple. Learning with privacy at scale. Technical report, Apple, 2017.

[4] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. The Privacy Blanket of the Shuffle Model. *arXiv e-prints*, art. arXiv:1903.02837, Mar 2019.

[5] Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 127–135. ACM, 2015.

[6] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, pages 441–459, 2017.

[7] Jaroslaw Błasiok, Mark Bun, Aleksandar Nikolov, and Thomas Steinke. Towards instance-optimal private query release. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2480–2497. SIAM, 2019.

[8] Mark Braverman, Ankit Garg, Tengyu Ma, Huy L. Nguyen, and David P. Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1011–1020. ACM, 2016.

[9] Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 435–447. ACM, 2018.

[10] Clément L. Canonne, Gautam Kamath, Audra McMillan, Adam Smith, and Jonathan Ullman. The structure of optimal private tests for simple hypotheses. *arXiv preprint arXiv:1811.11148*, 2018.

[11] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via mixnets. *arXiv preprint arXiv:1808.01394*, 2018.

[12] Rachel Cummings, Katrina Ligett, Kobbi Nissim, Aaron Roth, and Zhiwei Steven Wu. Adaptive learning with robust generalization guarantees. In *Conference on Learning Theory*, pages 772–814, 2016.

[13] Amit Daniely and Vitaly Feldman. Learning without interaction requires separation. *arXiv preprint arXiv:1809.09165*, 2018.

[14] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems*, pages 3574–3583, 2017.

[15] John Duchi and Ryan Rogers. Lower bounds for locally private estimation via communication complexity. *arXiv preprint arXiv:1902.00582*, 2019.

[16] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 429–438. IEEE, 2013.

[17] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

[18] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[19] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

[20] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM, 2019.

[21] Ankit Garg, Tengyu Ma, and Huy Nguyen. On communication cost of distributed statistical estimation and dimensionality. In *Advances in Neural Information Processing Systems*, pages 2726–2734, 2014.

[22] Svante Janson. Tail bounds for sums of geometric and exponential variables. *arXiv e-prints*, art. arXiv:1709.08157, Sep 2017.

[23] Matthew Joseph, Janardhan Kulkarni, Jieming Mao, and Zhiwei Steven Wu. Locally private gaussian estimation. *arXiv preprint arXiv:1811.08382*, 2018.

[24] Matthew Joseph, Aaron Roth, Jonathan Ullman, and Bo Waggoner. Local differential privacy for evolving data. In *Advances in Neural Information Processing Systems*, pages 2381–2390, 2018.

[25] Matthew Joseph, Jieming Mao, Seth Neel, and Aaron Roth. The role of interactivity in local differential privacy. *CoRR*, abs/1904.03564, 2019.

[26] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[27] Seth Neel, Aaron Roth, and Zhiwei Steven Wu. How to use heuristics for differential privacy. *arXiv preprint arXiv:1811.07765*, 2018.

[28] Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

[29] Ryan M. Rogers, Aaron Roth, Jonathan Ullman, and Salil Vadhan. Privacy odometers and filters: Pay-as-you-go composition. In *Advances in Neural Information Processing Systems*, pages 1921–1929, 2016.

[30] Adam Smith, Abhradeep Thakurta, and Jalaj Upadhyay. Is interaction necessary for distributed private learning? In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 58–77. IEEE, 2017.

[31] Jonathan Ullman. Tight lower bounds for locally differentially private selection. *arXiv preprint arXiv:1802.02638*, 2018.

[32] Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60 (309):63–69, 1965.