2012 IEEE 53rd Annual Symposium on Foundations of Computer Science

The Privacy of the Analyst and The Power of the State

Cynthia Dwork
Microsoft Research Silicon Valley
Mountain View, CA, USA
Email: dwork@microsoft.com

Moni Naor
Weizmann Institute of Science
Rehovot, Israel
Email: moni.naor@weizmann.ac.il

Salil Vadhan
Harvard University
Cambridge, MA
Email: salil@seas.harvard.edu

***Abstract*—We initiate the study of *privacy for the analyst* in differentially private data analysis. That is, not only will we be concerned with ensuring differential privacy for the data (i.e. individuals or customers), which are the usual concern of differential privacy, but we also consider (differential) privacy for the set of queries posed by each data analyst. The goal is to achieve privacy with respect to other analysts, or users of the system.**

This problem arises only in the context of *stateful* privacy mechanisms, in which the responses to queries depend on other queries posed (a recent wave of results in the area utilized cleverly coordinated noise and state in order to allow answering privately hugely many queries).

We argue that the problem is real by proving an exponential gap between the number of queries that can be answered (with non-trivial error) by stateless and stateful differentially private mechanisms. We then give a stateful algorithm for differentially private data analysis that also ensures differential privacy for the analyst and can answer exponentially many queries.

***Keywords*-differential privacy; list decoding; long code**

I. Introduction

Differential privacy is a widely studied notion of privacy designed for statistical analysis of confidential data [1], [2], [3]. All research on differential privacy to date has focused exclusively on the privacy of the data. In this work, we introduce the additional requirement of *privacy for the data analyst* — hiding the *questions* one asks about the data.

People studying a data set ("data analysts") may desire confidentiality for the questions they ask for a variety of reasons, including fear of embarrassment, persecution, leakage to competitors and in the case of law enforcement, informing criminals about the investigation. Allowing individuals to carry out research privately is well-recognized as being important for a free society, for example, as reflected in the strong protections given to library records. Thus, the Council of the American Library Association "*strongly recommends that the responsible officers of each library... Advise all librarians and library employees that such records shall not be made available to any agency of state, federal, or local government except pursuant to such process, order or subpoena*" [4]. Confidentiality of this type is also the motivation behind Private Information Retrieval [5], which ensures that the library itself does not learn what the user reads.

In this same spirit, we investigate the problem of *privacy for the data analyst* in differentially private mechanisms. Unlike in a library, or in PIR, where the information delivered to a user is exactly the information requested (the original, copyrighted, version of a book, for example), a feature of differential privacy is that the responses to queries suffer some (hopefully minor) distortion. In some algorithms, this distortion reveals information about queries posed to the system by other analysts, and thus may compromise their privacy. This is analogous to some of the concerns motivating secure computation of approximations, which ensure that approximations leak no more information than exact values [6], and history-independent data structures, which ensure that a data structure representation in memory is independent of the prior history of queries [7].

The absence of this natural question in the differential privacy literature may stem from several causes. First, the accuracy of responses to queries must deteriorate as the number and complexity of these queries increases ([8] *et sequelae*). Thus to obtain maximal utility of data it might make sense to publicize the answers to all the queries posed, so that the utility can be shared by all who desire this information. Second, the "first wave" of differentially private algorithms were all *stateless*, meaning that the probability distribution for the response to one query is the same regardless of what other queries have been asked (except perhaps with respect to the amount of the privacy budget the query is allowed to consume). Combining this with the fact that

Research supported in part by a grant the Israel Science Foundation and by a grant from the CITI Foundation. Part of this work was done while visiting Microsoft Research.

Center for Research on Computation and Society (CRCS) and School of Engineering and Applied Sciences (SEAS). Supported in part by a gift from Google, Inc. Work done in part while on leave as a Visiting Researcher at Microsoft Research SVC and a Visiting Scholar at Stanford University.

0272-5428/12 $26.00 © 2012 IEEE
DOI 10.1109/FOCS.2012.87

the curator is typically trusted,[1] the issue of privacy for the analyst was not investigated.

In a "second wave" of differentially private algorithms, initiated in [11] (see also [12], [13], [14], [15], [16], [17]), the responses to the different queries depend on other queries, either because the queries are handled as a batch [11], [14], [17], or because the algorithm explicitly maintains state [13], [15]. The benefit of these "second wave" algorithms is their ability to provide answers to truly huge numbers of queries, even exponential in the number of rows in the database (whereas the known stateless mechanisms can only handle up to a sub-quadratic number of queries).

The current work was motivated by seeking to distribute the work of the "second-wave" algorithms among multiple servers, and our subsequent realization that the need for shared state raises privacy concerns for the data analysts, even when the curator is trustworthy.

A. Our Results

State is Necessary: We first prove that a stateless differentially private algorithm cannot answer more than $\tilde{O}(n^2)$ counting queries[2] with nontrivial accuracy, where n is the number of rows in the database. This bound is tight up to polylogarithmic factors, and shows an exponential gap between the number of queries that can be answered by stateless and stateful differentially private mechanisms. The proof relies on the list-decoding properties of the "long code" [18].

Our result can be interpreted as a negative result about *distributing the work* of answering queries among servers while maintaining differential privacy: either the servers must share information about what queries are asked to them, or they can only answer a small number of queries. A second interpretation is that it may be difficult to achieve perfect privacy for the data analysts, if we have differential privacy for the dataset.[3]

Turning to the algorithmic problem of privacy for the analyst, we see immediately that this is impossible for a batch algorithm that takes a set of queries coming from different analysts and produces a public and differentially private summary of the dataset that allows each analyst to compute an accurate answer to its query: We know from lower bounds on noise [8] that the output of any batch algorithm must *fail* to accurately answer some potential queries. At the same time, with high probability all queries in the batch are answered accurately. Thus, any query not well answered is likely not to have been asked. We therefore consider only interactive mechanisms.

A Stateful Mechanism with Privacy for the Analysts: In light of our negative result, if we want privacy for both the data subjects and the data analysts, we must look at stateful algorithms. We will also settle for less than perfect privacy for the data analysts (e.g. look for an analogue of differential privacy). We achieve this in a model where every analyst is assigned an ID, which is fed to the mechanism along with every query made by that analyst. Specifically, we construct a stateful mechanism that:

- is differentially private for the data subjects
- can answer up to an exponential number of counting queries (as in the existing stateful mechanisms)
- provides analyst privacy in the following sense: the view of any one analyst (or few analysts) has approximately the same distribution regardless of what other queries are asked by *all* of the other analysts. Here "approximately the same distribution" is defined in the same sense as in the definition of differential privacy, namely every event occurs with the same probability up to a $(1+\varepsilon)$ multiplicative factor (and a negligible additive factor).

Our algorithm is based on nesting two privacy-preserving algorithms. The idea is to have two levels of responses — the inner layer and the outer layer. The inner layer is common to all the analysts and handles all their queries, without regard to which analyst issues which query. It answers them using a data-private algorithm, and guarantees that (whp) the accuracy of all the answers is within an additive $\alpha = \tilde{O}(1/n^{1/2})$ (hiding the dependence on all parameters other than the number n of rows in the database). We think of the inner layer as providing an α-accurate *oracle* for queries on the database. This layer is in charge of the privacy of the database elements. The second, outer, layer runs several instantiations of an algorithm, one for each analyst (the analyst is specified by an id). The purpose of the outer layer is to protect the privacy of the analysts: The algorithm does not operate on the real database, but views what the inner layer tells it as an α-accurate oracle for queries on the database, and its goal is to protect this oracle, that is to yield no information about *which* α-accurate oracle is used, since the precise nature of the oracle may yield information about the queries asked by the other analysts. The resulting scheme gives answers that are within $\pm\tilde{O}(\alpha^{1/2}) = \pm\tilde{O}(1/n^{1/4})$ of the real answer on all queries.

We remark that our analyst-private algorithm provides privacy for the *entire set* of queries posed by each

[1] Exceptions include [9] and all work in the *local model* [10].

[2] A counting query asks what fraction of the database lies in a specified subset of the universe $\mathcal{U}$.

[3] By perfect privacy for the data analysts, we mean that changing any query has no effect on the joint distribution of responses to the other queries. Stateless mechanisms provide this property, but are somewhat more constrained. We thank an anonymous reviewer for a question that brought out the difference.

analyst. This is analogous to a *user-level* privacy guarantee [19], [20]. The set of analysts, together, may make an exponential number of queries. In particular, any given user may make a truly huge number of queries, and the presence or absence of this entire, potentially huge, set is protected.

Additional Related Work: Kasiviswanathan *et al.*, in their work on differentially private release of contingency tables, consider a class of stateless mechanisms, which they call *instance-independent*, obtaining stronger lower bounds on distortion for these mechanisms than they obtain for general mechanism [21].

Subsequent to the current paper, Ullman [22] proved that, under standard cryptographic assumptions, it is computationally intractable for even a stateful differentially private mechanism to answer more than $n^{2+o(1)}$ arbitrary (but efficiently computable) counting queries on a large data universe. His proof exploits an intimate connection to "traitor-tracing schemes" from [12] and uses some ideas related to our negative result. Intuitively, he constructs an adversarial sequence of counting queries that can be viewed as encrypted versions of the queries in our attack, so that it is infeasible for the mechanism to take advantage of the correlations between the queries despite being stateful.

II. The Power of State

In this section, we prove that a stateless differentially private algorithm cannot answer more than a quadratic number of (counting) queries with nontrivial accuracy. We prove our negative result for statelessness in the "easiest possible" case for a stateless algorithm: we have a large set of processors, each of which will respond to at most a single query. The processors share the same database, and they may have an unlimited amount of shared *initial* state, including an arbitrarily long random tape. At each step of the attack, the adversary chooses a processor, poses a query to this processor, and receives a response. The processors have unrestricted computational resources, and each processor may have its own program, say, depending on its processor id[4]. However, the processors do not communicate once interaction with the data analyst (the adversary) has begun[5]. Thus, in this work *statefulness* and *coordination* are equivalent.

[4]A more stringent requirement is *order-obliviousness*, in which the answer to the ith query does not depend on i. Our adversary succeeds even with the less stringent requirement described above.

[5]What makes this the "easiest possible" case for a stateless differentially private algorithm (with non-trivial accuracy) is the sharing of *initial* state and randomness. If there is no sharing of private randomness then a fairly straightforward hybrid argument and sampling argument show that differential privacy can be defeated with $\omega(n^2)$ queries to the noncommunicating processors. This also has implications for cumulative privacy loss over participation in multiple, independently operated, databases.

We consider databases that consist of n uniformly random rows from some data universe $\mathcal{U}$ (chosen without replacement), and mechanisms that answer counting queries. For any stateless mechanism that provides a nontrivial bound on the expected error, we exhibit an efficient adversary that makes $O(n^2 \log |\mathcal{U}|)$ counting queries and can reconstruct an unknown row of the database with probability $\Omega(1/n^2)$ (based on knowing the other $n-1$ rows of the database). This implies that the mechanism cannot be differentially private, provided that the data universe is of size $|\mathcal{U}| = \omega(n^2)$ (which is also tight — see Section II-B).

Several remarks are in order:

(1) The success probability of the attack is independent of the size of the universe $\mathcal{U}$, although the number of queries needed to launch the attack depends logarithmically on $|\mathcal{U}|$. For a small universe, this probability of success can be achieved simply by guessing a random member of the universe (even without posing any queries), so the result is significant only when the universe has size $\omega(n^2)$ (and we contradict differential privacy only in this case). If we think of a database as containing the data of n different Americans, identified by their social security numbers, then the universe is at least the size of the US population, and the reconstruction attack will produce the social security number of a member of the database, together with the rest of this individual's data. If, moreover, this is a database, say, of HIV-positive individuals, then the attack immediately identifies someone as being HIV-positive.

(2) Suppose each datum is very complex; for example, it may be a patient's name together with his completely sequenced DNA. In this case the universe is huge, but the attack is flexible in that it permits the attacker to focus on, say, 100 "interesting" bits in the DNA sequence. In this case a successful adversary produces the name of the individual together with these 100 bits.

A. Model for Proving the Separation Result

The database is a collection of elements, each drawn from a universe $\mathcal{U}$, and queries map databases to reals. In this section, our databases are (unordered) sets (not multisets), and queries operate on subsets of the universe $\mathcal{U}$ of elements.

A *query* is a function $q : 2^{\mathcal{U}} \to \mathbb{R}$. For a data universe $\mathcal{U}$, an integer $t \in \mathcal{N}$, and a query family $\mathcal{Q} = \{q : 2^{\mathcal{U}} \to \mathbb{R}\}$, a *query release mechanism* is a randomized function $M : 2^{\mathcal{U}} \times \mathcal{Q}^t \to \mathbb{R}^t$, which takes a database $x \in 2^{\mathcal{U}}$ and a sequence of t queries $q^{(1)}, \ldots, q^{(t)} \in \mathcal{Q}$, and outputs a sequence $M(x, q^{(1)}, \ldots, q^{(t)}) = (y^{(1)}, \ldots, y^{(t)})$ where $y^{(j)}$ is intended to be an estimate of $q^{(j)}(x)$. When we want to make the coin tosses r of M explicit, we will write $M(x, q^{(1)}, \ldots, q^{(t)}; r)$.

Databases $x, x' \in 2^{\mathcal{U}}$ are *adjacent* if they satisfy $|x \Delta x'| \leq 1$.

Definition II.1. Random variables Y and Z are (ε, δ)-indistinguishable if for every set S, we have

$$\begin{aligned} \Pr[Y \in S] &\leq \exp(\varepsilon) \cdot \Pr[Z \in S] + \delta, \text{ and} \\ \Pr[Z \in S] &\leq \exp(\varepsilon) \cdot \Pr[Y \in S] + \delta. \end{aligned}$$

Definition II.2. A query release mechanism $M : 2^{\mathcal{U}} \times \mathcal{Q}^t \to \mathbb{R}^t$ is (ε, δ) *differentially private* iff for all adjacent databases $x, x' \in 2^{\mathcal{U}}$ and all query sequences $q^{(1)}, \ldots, q^{(t)} \in \mathcal{Q}$, the random variables $M(x, q^{(1)}, \ldots, q^{(t)})$ and $M(x', q^{(1)}, \ldots, q^{(t)})$ are (ε, δ)-indistinguishable (over the coin tosses of M).

Typically, we think of ε as a small constant, and δ as negligibly small (e.g. $\delta = 1/n^{\omega(1)}$). The above definition only considers privacy for nonadaptive queries, making our negative result stronger. For our positive result in Section III, we achieve privacy even for adaptive queries.

Definition II.3. We say that a query release mechanism $M : 2^{\mathcal{U}} \times \mathcal{Q}^t \to \mathbb{R}$ is *stateless* iff for every $j \in [t]$, the answer to the jth query does not depend on the other $t-1$ queries given to M; *i.e.*,

$$\begin{aligned} &M(x, q^{(1)}, \ldots, q^{(t)}; r) \\ &\quad = \quad (M^{(1)}(x, q^{(1)}; r), \ldots, M^{(t)}(x, q^{(t)}; r)) \end{aligned}$$

for some mechanisms $M^{(1)}, \ldots, M^{(t)}$.

We now define a game played by the adversary where a 'win' for the adversary is a privacy compromise. We begin with an informal description: A random database x is chosen and the adversary is given all but one element ξ from the database; such an adversary is sometimes referred to as "totally informed". Based on $x \setminus \xi$, the adversary then asks some t queries to the mechanism, and tries to guess (or"reidentify") the unknown element ξ of the database. The adversary wins if it guesses correctly.

In order to obtain tighter parameters in our negative results about differential privacy, we will consider a generalization of the above game where the adversary instead outputs a probability distribution p on the data universe, where $p(w)$ represents the adversary's confidence that $\xi = w$. Thus, if the adversary guessed the value of ξ according to p, the probability of reidentification would be $p(\xi)$. However, we will instead give the adversary a payoff of $\sqrt{p(\xi)}$. This can be related to reidentification probability by the relation $\mathrm{E}[p(\xi)] \geq \mathrm{E}[\sqrt{p(\xi)}]^2$, but will enable tighter lower bounds for differential privacy than analyzing only $\mathrm{E}[p(\xi)]$.

Definition II.4. Let $M : 2^{\mathcal{U}} \times \mathcal{Q}^t \to \mathbb{R}^t$ be a query-release mechanism and let $n \in \mathcal{N}$ be a database size. For a (randomized and computationally unbounded) adversary A, the *totally informed reidentification game* is defined as follows:

1. Let x be a uniformly random subset of $\mathcal{U}$ of size n.
2. Let ξ be a uniformly random element of x.
3. Feed the set $\xi^c \stackrel{\text{def}}{=} x \setminus \xi$ to A, who then outputs a query sequence $q^{(1)}, \ldots, q^{(t)}$.
4. Run $M(x, q^{(1)}, \ldots, q^{(t)})$ to obtain output $y = (y^{(1)}, \ldots, y^{(t)})$.
5. Feed y to A, who then outputs a probability distribution p on $\mathcal{U}$.

A's *payoff* is defined to be $\sqrt{p(\xi)}$. The expected value of A's payoff, $\mathrm{E}[\sqrt{p(\xi)}]$, is over all the randomness in the above game (including the randomness of both M and A).

Proposition II.5. *If $M : 2^{\mathcal{U}} \times \mathcal{Q}^t \to \mathbb{R}^t$ is an (ε, δ)-differentially private query release mechanism, then for every (randomized and computationally unbounded) adversary A: A's expected payoff in the totally informed reidentification game is at most $e^{\varepsilon} \cdot \sqrt{1/(|\mathcal{U}| - (n-1))} + \delta$.*

Proof: By (ε, δ) differential privacy, A's expected payoff is at most $e^{\varepsilon} \cdot \mu + \delta$, where μ is A's expected payoff in a modified game where we feed the mechanism M only ξ^c rather than all of x. In this modified game, ξ is equally likely to be any element of $\mathcal{U} \setminus \xi^c$ even conditioned on A's view. Thus, when A outputs probability distribution p, its expected payoff is

$$\mathrm{E}_{\xi \leftarrow \mathcal{U} \setminus \xi^c}\left[\sqrt{p(\xi)}\right] \leq \sqrt{\mathrm{E}_{\xi \leftarrow \mathcal{U} \setminus \xi^c}\left[p(\xi)\right]} \leq \sqrt{\frac{1}{|\mathcal{U}| - (n-1)}}.$$

■

We will use the following measure of utility in our negative result.

Definition II.6. We say that a query release mechanism $M : 2^{\mathcal{U}} \times \mathcal{Q}^t \to \mathbb{R}^t$ has *expected error* γ if for every database $x \in 2^{\mathcal{U}}$, sequence $q^{(1)}, \ldots, q^{(t)} \in \mathcal{Q}$, and $j \in [t]$, if we let $(y^{(1)}, \ldots, y^{(t)}) \leftarrow M(x, q^{(1)}, \ldots, q^{(t)})$, we have

$$\mathrm{E}[|y^{(j)} - q^{(j)}(x)|] \leq \gamma.$$

The probability space is over the coin flips of the adversary and the mechanism.

Counting Queries: In the literature, a (fractional) *counting query* is specified by a predicate $q : \mathcal{U} \to \{0, 1\}$. When evaluated on a database $x \subseteq \mathcal{U}$, the counting query q gives the fraction of elements of x that satisfy the predicate. We abuse notation and write q to denote both the predicate on $\mathcal{U}$, and the corresponding function on databases (which are subsets

of $\mathcal{U}$). Specifically, for a predicate $q : \mathcal{U} \to \{0,1\}$ and a database $x \subseteq \mathcal{U}$ of size n, we have:

$$q(x) = \frac{1}{|x|}\sum_{w\in x} q(w) = \frac{1}{n}\sum_{w\in x} q(w).$$

For technical convenience, in this section we formulate counting queries as $\{\pm 1\}$-valued predicates; That is, $q : \mathcal{U} \to \{\pm 1\}$, so that $q(x) = \frac{1}{|x|}\sum_{z\in x} q(z) = \frac{1}{n}\sum_{z\in x} q(z) \in [-1,1]$.

B. The Separation Result

Our main negative result is given by the following attack on stateless mechanisms:

Theorem II.7. *There is a universal constant c such that the following holds. Let $\mathcal{Q} = \{\pm 1\}^{|\mathcal{U}|}$ be the set of all counting queries on data universe $\mathcal{U}$, and let $M : 2^{\mathcal{U}} \times \mathcal{Q}^t \to [-1,1]^t$ be a stateless query release mechanism that has expected error at most $1-\gamma$ and supports $t \geq cn^2 \log|\mathcal{U}|$ queries. Then there is an adversary, running in time $\mathrm{poly}(n,t,|\mathcal{U}|)$ that achieves payoff $\Omega(\gamma/n)$ in the totally informed reidentification game. In particular, if $\gamma = \Omega(1)$, $|\mathcal{U}| = \omega(n^2)$, $\varepsilon = O(1)$, and $\delta = o(1/n)$, them M cannot be (ε,δ)-differentially private.*

This theorem is nearly tight in almost all parameters: The requirement that $t \gtrsim n^2$ is necessary because $o(n^2)$ queries can be privately answered using independent noise. The condition that $|\mathcal{U}| \gtrsim n^2$ is necessary, because for data universes of size $o(n^2)$, many counting queries can be answered using "randomized response". Requiring $\delta \lesssim 1/n$ is necessary because random subsampling achieves $(0, \tilde{O}(1/n))$ differential privacy and can answer many queries accurately. And providing the adversary some information about the database is necessary because otherwise a simple "density estimation" strategy can compute an accurate response with high probability without even looking at the database (thereby providing perfect privacy).

Proof of Theorem II.7: As in the totally informed reidentification game, we consider a database x that is a uniformly random set of n distinct elements drawn from $\mathcal{U}$, and we write $x = (\xi, \xi^c)$, where ξ^c is the set of elements in x known to the adversary. The adversary, $A(\xi^c)$, generates its counting queries $q^{(1)},\ldots,q^{(t)} \in \mathcal{Q}$ as follows.

1. Choose $k \leftarrow \{0,1,\ldots,n-1\}$.
2. Choose a uniformly random predicate $q_\bullet$ (pronounced "q-known), where $q_\bullet : \xi^c \to \{\pm 1\}$, such that $q_\bullet(w) = 1$ for exactly k elements $w \in \xi^c$, so $q_\bullet$ has value 1 on exactly k elements known by the adversary to be in the database and value -1 on the remaining $n-1-k$ elements known by the adversary to be in the database.
3. For $j = 1,\ldots,t$:

   Select $q_\circ^{(j)} : (\mathcal{U}\setminus\xi^c) \to \{\pm 1\}$ (pronounced "q-unknown") uniformly at random, so elements *not* known by the adversary to be in the database are included in the query with probability $1/2$.

   Let $q^{(j)} = (q_\bullet, q_\circ^{(j)}) : \mathcal{U} \to \{\pm 1\}$ be the predicate that equals $q_\bullet$ on ξ^c and equals $q_\circ^{(j)}$ on $\mathcal{U}\setminus\xi^c$.
4. Output the queries $(q^{(1)},\ldots,q^{(t)})$.

The attack is not adaptive, strengthening the result.

Upon receiving the response $(y^{(1)},\ldots,y^{(t)}) = (M^{(1)}(x,q^{(1)};r),\ldots,M^{(t)}(x,q^{(t)};r))$, the adversary computes, for each element $w \in \mathcal{U}\setminus\xi^c$:

$$c(w) = \frac{1}{t}\sum_{j=1}^{t} q_\circ^{(j)}(w)\cdot y^{(j)}$$

and outputs any probability distribution that assigns each element $w \in \mathcal{U}\setminus\xi^c$ probability at least $p(w) = \max\{c(w) - \gamma/2n, 0\}^2$. (If $\sum_w p(w) > 1$, the adversary fails and the payoff is 0.)

We now provide some intuition for the analysis of the expected payoff. Consider any fixed setting of $x = (\xi,\xi^c)$, k, r, and $q_\bullet$. Conditioned on these values the expectation of $c(w)$ is the correlation between the function $g : \{\pm 1\}^{\mathcal{U}\setminus\xi^c} \to [-1,1]$ defined as

$$g(q_\circ) = \mathop{\mathrm{E}}_j[M^{(j)}(x,(q_\bullet,q_\circ);r)]$$

and the "dictator" function χ_w that maps each $q_\circ \in \{\pm 1\}^{\mathcal{U}\setminus\xi^c}$ to $q_\circ(w)$. (Note that χ_w is the encoding of w in the *long code* of [18].) That is, conditioned on these values of x, k, and $q_\bullet$, we have

$$\begin{aligned}\mathrm{E}[c(w)] &= \langle g, \chi_w\rangle \\ &= \frac{1}{2^{|\mathcal{U}|-n+1}}(\text{dot product of } g \text{ and } \chi_w \text{ as vectors}) \\ &= \mathrm{E}_{q_\circ}[g(q_\circ)\chi_w(q_\circ)].\end{aligned}$$

Thus, by setting $p(w) = \max\{c(w) - \gamma/2n, 0\}^2$, our adversary is trying to identify and assign positive probability to all w such that g has significant correlation with the dictator function χ_w. This can be viewed as the task of "list-decoding" g according to the long code.

If M were to always answer with perfect accuracy (without adding noise for privacy), then its outputs would be completely uncorrelated with all dictator functions except that of ξ. However, all the adversary has to work with is that, in some average sense, the mechanism's responses must be correlated with (most) elements in the database, and independent of (most) elements outside of the database.

We next analyze the expected payoff of the adversary, assuming that $\sum_w p(w) > 1$ does not occur.

Claim II.8.

$$\mathrm{E}\left[\sqrt{p(\xi)}\right] \geq \frac{\gamma}{2n},$$

where the expectation is taken above all the randomness in the above experiment (namely x, k, r, $q_\bullet$, and $q_\circ^{(1)}, \ldots, q_\circ^{(t)}$).

Proof: We have

$$\mathrm{E}\left[\sqrt{p(\xi)}\right] = \mathrm{E}\left[\max\{c(\xi) - \gamma/2n, 0\}\right] \geq \mathrm{E}[c(\xi)] - \gamma/2n.$$

Thus it suffices to show $\mathrm{E}[c(\xi)] \geq \gamma/n$. First, note that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{x,k,r,q_\bullet,q_\circ^{(1)},\ldots,q_\circ^{(t)}}[c(\xi)] \\
&= \mathop{\mathrm{E}}_{x,k,r,q_\bullet,q_\circ^{(1)},\ldots,q_\circ^{(t)}}\left[\frac{1}{t}\sum_{j=1}^{t} q_\circ^{(j)}(\xi)\cdot M^{(j)}(x,q^{(j)};r)\right] \\
&= \mathop{\mathrm{E}}_{j,x,k,r,q_\bullet,q_\circ}\left[q_\circ(\xi)\cdot M^{(j)}(x,(q_\bullet,q_\circ);r)\right],
\end{aligned}$$

where the first equality is by definition and in the last expression $j \leftarrow [t]$ and $q_\circ$ is a uniformly random function from $\mathcal{U}\setminus\xi^c$ to $\{\pm 1\}$.

With this simplified notation, we proceed as follows:

$$\begin{aligned}
\mathrm{E}[c(\xi)] &= \mathop{\mathrm{E}}_{j,x,k,r,q_\bullet,q_\circ}\left[q_\circ(\xi)\cdot M^{(j)}(x,(q_\bullet,q_\circ);r)\right] \\
&= \left(\frac{1}{2}\right)\mathop{\mathrm{E}}_{j,x,k,r,q_\bullet}\left[\mathop{\mathrm{E}}_{q_\circ:q_\circ(\xi)=1}[M^{(j)}(x,(q_\bullet,q_\circ);r)]\right. \\
&\quad \left. - \mathop{\mathrm{E}}_{q_\circ:q_\circ(\xi)=-1}[M^{(j)}(x,(q_\bullet,q_\circ);r)]\right] \\
&= \left(\frac{1}{2}\right)\mathop{\mathrm{E}}_{j,x,k,r}\left[\mathop{\mathrm{E}}_{q:\#\{w\in x:q(w)=1\}=k+1}[M^{(j)}(x,q;r)]\right. \\
&\quad \left. - \mathop{\mathrm{E}}_{q:\#\{w\in x:q(w)=1\}=k}[M^{(j)}(x,q;r)]\right] \\
&= \left(\frac{1}{2n}\right)\mathop{\mathrm{E}}_{j,x,r}\left[\mathop{\mathrm{E}}_{q:\#\{w\in x:q(w)=1\}=n}[M^{(j)}(x,q;r)]\right. \\
&\quad \left. - \mathop{\mathrm{E}}_{q:\#\{w\in x:q(w)=1\}=0}[M^{(j)}(x,q;r)]\right] \\
&\geq \left(\frac{1}{2n}\right)\cdot(1-(1-\gamma)-(-1+(1-\gamma))) = \frac{\gamma}{n}
\end{aligned}$$

■

The next claim shows that the event $\sum_w p(w) > 1$ occurs rarely, and thus has little effect on $\mathrm{E}[\sqrt{p(\xi)}]$.

Claim II.9. *With probability at least $1-\gamma/4n$, we have $\sum_w p(w) \leq 1$.*

Proof: Consider any fixed setting of $x = (\xi, \xi^c)$, k, r, and $q_\bullet$. As in the proof of Claim II.8, the expectation of $c(w)$ conditioned on these values is exactly:

$$\hat{g}(w) = \mathop{\mathrm{E}}_{j,q_\circ}[q_\circ(w)M^{(j)}(x,(q_\bullet,q_\circ);r)].$$

and recall that $\hat{g}(w)$ is exactly the correlation between $g : \{\pm 1\}^{\mathcal{U}\setminus\xi^c} \to [-1,1]$ defined by

$$g(q_\circ) = \mathop{\mathrm{E}}_{j}[M^{(j)}(x,(q_\bullet,q_\circ);r)]$$

and the dictator function that maps each $q_\circ \in \{\pm 1\}^{\mathcal{U}\setminus\xi}$ to $q_\circ(w)$. Dictators constitute the first level of the Fourier basis over $\{\pm 1\}^m$). By Parseval's Identity, we have

$$\sum_w \hat{g}(w)^2 \leq \mathop{\mathrm{E}}_{q_\circ}[g(q_\circ)^2] \leq 1.$$

(Parseval's Identity becomes an inequality here because the dictators are only a subset of the Fourier basis.)

To show that $\sum_w c(w)^2$ is also bounded with high probability, we observe that each $c(w)$ is the average of the t random variables $q_\circ^{(j)}(w)\cdot M^{(j)}(x,(q_\bullet,q_\circ^{(j)});r) \in [-1,1]$, which are independent once we fix $x = (\xi,\xi^c)$, k, r, and $q_\bullet$. Thus, by a Chernoff bound and union bound, the probability that $c(w) > \hat{g}(w) + \gamma/2n$ for some w is at most $|\mathcal{U}|\cdot\exp(-\Omega(t\cdot(\gamma/2n)^2)) \leq \gamma/4n$ by the choice of $t \geq c\cdot(n/\gamma)^2\cdot\log|\mathcal{U}|$. (We may assume that $|\mathcal{U}| \geq (n/\gamma)^2$, else the adversary can achieve payoff γ/n by just outputting the uniform distribution on $\mathcal{U}$.) As long as $c(w) \leq \hat{g}(w) + \gamma/2n$ for all w, we have

$$\sum_w p(w) = \sum_w \max\{c(w) - \gamma/2n, 0\}^2 \leq \sum_w \hat{g}(w)^2 \leq 1.$$

■

By Claims II.8 and II.9, the expected payoff of our adversary is at least:

$$\mathrm{E}[\sqrt{p(\xi)}] - \Pr\left[\sum_w p(w) \geq 1\right] \geq \gamma/2n - \gamma/4n = \Omega(\gamma/n).$$

■

Remarks and Extensions: The above proof only requires a very weak consequence of expected error, namely that on a uniformly random database $x \subseteq \mathcal{U}$ of size n, a uniformly random counting query q that is constant on the rows of x, and a uniformly random $j \in [t]$, the expectation of $M^{(j)}(x,q)$ is within $\pm(1-\gamma)$ of $q(x)$ (which is either 1 or -1).

In case the expected error is a relatively small α (instead of being close to 1), the adversary can attack with knowledge of substantially fewer rows. Suppose the adversary knows some number $n' - 1 < n$ rows of x, together with a bound α on the expected error. Writing $n' = \lceil(\alpha+\gamma)n\rceil$, and solving for γ, the adversary can launch of modification of the attack, with improved expected payoff $\Omega((\gamma/n)) = \Omega(\gamma/((\gamma+\alpha)n))$.

When the data universe is large, the number t of queries needed by our adversary grows proportionally to $\log|\mathcal{U}|$, and the description size of a query and running time of our adversary grow proportionally to $|\mathcal{U}|$. These blow-ups can be remedied by effectively reducing the

universe size, either by considering databases where the rows come from a smaller subset $\mathcal{U}' \subseteq \mathcal{U}$, or by considering an adversary that only tries to learn the first few "attributes" of a row (i.e. take $\mathcal{U} = \mathcal{U}' \times \mathcal{U}''$, consider queries that only look at the $\mathcal{U}'$ component, and construct an adversary that outputs the $\mathcal{U}'$ component of a random row). Restricting the data universe in either of these ways preserves differential privacy.

To have an adversary that learns all $\log|\mathcal{U}|$ bits of a row chosen uniformly from $\mathcal{U}$, then the number of queries must grow proportionally to $\log|\mathcal{U}|$ by information-theoretic arguments. However, the description size of queries can be reduced from $|\mathcal{U}|$ to $O(n \log|\mathcal{U}|)$ by using counting queries whose underlying predicates come from an $(n+1)$-wise independent family $\mathcal{Q}$ of hash functions $q : \mathcal{U} \to \{\pm 1\}$.

Using $(n+1)$-wise independent hash functions as in the previous item, we can also reduce the running time of the adversary to $\mathrm{poly}(n, \log|\mathcal{U}|)$, while still having the adversary learn all $\log|\mathcal{U}|$ bits of information about a uniformly random row. Specifically, we use a family $\mathcal{Q}$ of hash functions where each $q \in \mathcal{Q}$ is described a bit string $\tilde{q}$ of length $m = O(n \log|\mathcal{U}|)$, and where for every $w \in \mathcal{U}$, $q(w)$ is an $\mathbb{F}_2$-linear function of $\tilde{q}$. That is, $q(w) = (-1)^{\langle \tilde{q}, \ell_w \rangle}$ for some bit-string ℓ_w of length m.

We exploit this linear structure in the adversary as follows. The adversary selects $q_\bullet : \xi^c \to \{\pm 1\}$ as in the current attack. Restricting the query $q^{(j)}$ to agree with $q_\bullet$ on ξ^c amounts to imposing $n-1$ $\mathbb{F}_2$-linear constraints on the description of $q^{(j)}$. That is, we can now describe each query $q_\circ^{(j)}$ by a bitstring $\tilde{q}_\circ^{(j)}$ of length $m - n + 1$, and for every $w \notin \xi^c$, we have $q_\circ^{(j)}(w) = (-1)^{\langle \tilde{q}_\circ^{(j)}, \ell'_w \rangle}$, where ℓ'_w is also of length $m - n + 1$. Now we can efficiently find all w such that $g(q_\circ)$ has noticeable correlation with the dictator function $\chi_w(q_\circ) = q_\circ(w) = (-1)^{\langle \tilde{q}_\circ, \ell'_w \rangle}$ using the Goldreich–Levin algorithm [23].

III. Privacy for the Analyst

We now show that it is possible for a centralized curator to give rigorous guarantees on the privacy of the analysts while maintaining differential privacy for the data subjects against exponentially many queries. Our mechanism will be stateful (as is necessary by Theorem II.7), and will also require assigning analysts IDs (see below). The curator will ensure that the coordination of answers does not leak substantial information about the queries. This is done by yet another level of coordination.

A. Model and Definitions

Definition III.1 (stateful mechanisms with IDs)**.** For a data universe $\mathcal{U}$, an integer $t \in \mathcal{N}$, a query family $\mathcal{Q} = \{q : 2^{\mathcal{U}} \to \mathbb{R}\}$, and an ID space $\mathcal{I}$, a *stateful query mechanism with analyst IDs* is a randomized function $M : 2^{\mathcal{U}} \times \mathcal{Q} \times \mathcal{I} \times \mathcal{S} \to \mathbb{R} \times \mathcal{S}$, which takes a database $x \in 2^{\mathcal{U}}$, a query $q \in \mathcal{Q}$, an analyst $\mathrm{id} \in \mathcal{I}$, and a state $s \in \mathcal{S}$ and outputs an answer $y \in \mathbb{R}$ (intended to be an approximation of $q(x)$ and a new state s'. M also comes associated with an initial state $s_0 \in \mathcal{S}$. If $\mathcal{I} = \emptyset$, we simply refer to M as a *stateful query mechanism* (without analyst IDs).

We will require and achieve privacy even against adversaries that ask their queries adaptively (in contrast to Definition II.2 used in our negative result). Recall Definition II.1 of (ε, δ)-indistinguishable.

Definition III.2 (differential privacy for stateful mechanisms)**.** Let $M : 2^{\mathcal{U}} \times \mathcal{Q} \times \mathcal{I} \times \mathcal{S} \to \mathbb{R} \times \mathcal{S}$ be a stateful query mechanism with analyst IDs. We say that M is (ε, δ) *differentially private* if the following holds for every two adjacent databases $x, x' \in 2^{\mathcal{U}}$ and every randomized adversary A that adaptively queries M (i.e. submits a (query, id) pair $(q^{(1)}, \mathrm{id}^{(1)})$, receives a response $y^{(1)}$, then computes its next query $(q^{(2)}, \mathrm{id}^{(2)})$, and so on): the view of A (consisting of the coins of A and all the responses $y^{(i)}$) when interacting with $M(x, \cdot, \cdot, \cdot)$ and the view of A when interacting with $M(x', \cdot, \cdot, \cdot)$ are (ε, δ)-indistinguishable.

We also require accuracy when the queries are posed adaptively.

Definition III.3 (accuracy for stateful mechanisms)**.** Let $M : 2^{\mathcal{U}} \times \mathcal{Q} \times \mathcal{I} \times \mathcal{S} \to \mathbb{R} \times \mathcal{S}$ be a stateful query mechanism with analyst IDs. We say that M is (α, β) *accurate for t queries on databases of size n* if for every database $x \in 2^{\mathcal{U}}$ of size n and every every randomized adversary A that adaptively queries M with queries $q^{(1)}, \ldots, q^{(t)}$, with probability at least $1 - \beta$, all the responses $y^{(j)}$ differ from $q^{(j)}(x)$ by at most α.

Now we define privacy for the analyst. This definition will rely on the analyst IDs, and guarantees that no analysts can learn much about the other analysts' queries.

Definition III.4 (analyst privacy for stateful mechanisms)**.** Let $M : 2^{\mathcal{U}} \times \mathcal{Q} \times \mathcal{I} \times \mathcal{S} \to \mathbb{R} \times \mathcal{S}$ be a stateful query mechanism with analyst IDs. We say that M provides (ε, δ) *many-to-one analyst privacy* if for every database $x \in 2^{\mathcal{U}}$, every $\mathrm{id} \in \mathcal{I}$, and every two randomized, adaptive "honest" query strategies H_0 and H_1 that can issue queries with any IDs other than id, and every randomized, adaptive adversary A that always issues queries under id, the views of A in the following experiment when $H = H_0$ and $H = H_1$ are (ε, δ)-indistinguishable:

A and H interact in an interleaved manner with $M(x,\cdot,\cdot,\cdot)$, where A determines when H gets to make queries and how many queries H can make (but does not see the queries made or the results of those queries).

B. The Analyst-Private Mechanism

We show how to construct a mechanism that satisfies the above notion of privacy for the analyst, differential privacy for the data subjects, and provides accuracy for a large number of queries. This is summarized in the following theorem:

Theorem III.5. *Let $\mathcal{Q} = \{0,1\}^{\mathcal{U}}$ be the set of all counting queries on data universe $\mathcal{U}$, let $\beta, \delta, \varepsilon, \varepsilon' \in (0,1)$, and let $t \in \mathcal{N}$. There is a stateful mechanism with analyst IDs that:*

- *Is (ε, δ) differentially private,*
- *Provides (ε', β) many-to-one privacy for the analysts, and*
- *Is (α, β) accurate for up to t queries on databases of size n, where α is*

$$O\left(\frac{\left(\log^{3/8}|\mathcal{U}|\right)\left(\log^{1/8}(1/\delta)\right)\left(\log^{1/4}(1/\beta)\right)\left(\log^{3/4}(t/\beta)\right)}{(\varepsilon n)^{1/4}\cdot(\varepsilon')^{1/2}}\right).$$

Note that we can take $1/\delta$, $1/\beta$, $|\mathcal{U}|$, and t to all be $2^{n^{\Omega(1)}}$, and ε, ε' to be $1/n^{\Omega(1)}$ and still have the error vanishing polynomially in n. Also note that the many-to-one privacy for the analysts is guaranteed with β rather than its own separate parameter δ', since there is a tight connection between the accuracy guarantee of the inner one and the privacy provided for the analyst.

Our algorithm is based on a nested version of the Private Multiplicative Weights (PMW) algorithm of Hardt and Rothblum [15], which achieves the best parameters of any known stateful differentially private algorithm (using its analysis from [24]; a simpler proof appears in [25]):

Theorem III.6 (Private Multiplicative Weights [15], [24])**.** *Let $\mathcal{Q} = \{0,1\}^{\mathcal{U}}$ be the set of all counting queries on data universe $\mathcal{U}$, let $\beta, \delta, \varepsilon, \varepsilon' > 0$ be real numbers, and let $t \in \mathcal{N}$. There is a stateful mechanism that:*

- *Is (ε, δ) differentially private,*
- *Is (α, β) accurate on for up to t queries on databases of size n, where α is*

$$O\left(\frac{\left(\log^{1/4}|\mathcal{U}|\right)\left(\log^{1/4}(1/\delta)\right)\left(\log^{1/2}(t/\beta)\right)}{(\varepsilon n)^{1/2}}\right).$$

Our algorithm utilizes a single, long-lived, "inner" instantiation of the PMW algorithm, denoted $\mathrm{PMW}_{\mathrm{inner}}$, which provides privacy for the data subjects over all the queries posed by all the analysts. For this part we simply rely on the standard differential privacy properties of the algorithm and not on any specific characteristics of it (i.e. any algorithm with good differential privacy would do).

Then, for each analyst (as specified by their id), we spawn an "outer" instantiation of the PMW algorithm, denoted $\mathrm{PMW}_{\mathrm{id}}$. These outer $\mathrm{PMW}_{\mathrm{id}}$ algorithms do not access the database directly, but only through $\mathrm{PMW}_{\mathrm{inner}}$. To show that an analyst with a given id does not learn much about the queries of the other analysts, we combine the *accuracy* properties of $\mathrm{PMW}_{\mathrm{inner}}$ with the *privacy* properties of $\mathrm{PMW}_{\mathrm{id}}$. Specifically, regardless of what questions are asked by the other analysts, $\mathrm{PMW}_{\mathrm{inner}}$ will still respond to $\mathrm{PMW}_{\mathrm{id}}$ with answers that are within $\pm\alpha$ of what the database itself would have provided (except with probability at most β).

By generalizing the PMW privacy analysis, we show that the output distribution of $\mathrm{PMW}_{\mathrm{id}}$ is approximately the same as it would be if it accessed the database directly (instead of through $\mathrm{PMW}_{\mathrm{inner}}$). Indeed, we show that this holds not just for accessing the database through $\mathrm{PMW}_{\mathrm{inner}}$ but through any stateful oracle that provides answers that are close to those obtained on the true database. Consider the following definition of oracle-aided mechanism and the corresponding privacy requirements. The intuition is that the value given by the oracle is taken "as truth." The oracle may be stateful, and does not have to give consistent answers (repeating a query need not yield the same result).

Definition III.7 (oracle-aided mechanism)**.** A stateful mechanism $M : \mathcal{Q} \times \mathcal{S} \to \mathbb{R} \times \mathcal{S}$ is said to be an oracle-aided stateful query mechanism if to answer queries $q \in \mathcal{Q}$ on a database $x \in 2^{\mathcal{U}}$, it never directly accesses the database, but instead forwards q to a (possibly stateful) "oracle" $\hat{x} : \mathcal{Q} \times \mathcal{S}' \to \mathbb{R} \times \mathcal{S}'$, where $\mathcal{S}'$ is the set of states of the oracle, and uses the oracle's answer together with its (the mechanism's) state to respond.

One example of a possible oracle $\hat{x}$ is the database x itself, which does not use any state and responds to any query q with $q(x)$.

Definition III.8 ((ε, δ) privacy for α-accurate oracles)**.** Let $M : \mathcal{Q} \times \mathcal{S} \to \mathbb{R} \times \mathcal{S}$ be an oracle-aided stateful query mechanism. We say that M has (ε, δ) *privacy for α-accurate oracles* if the following holds: for every database $x \in 2^{\mathcal{U}}$, every stateful oracle $\hat{x} : \mathcal{Q} \times \mathcal{S} \to \mathbb{R}$ that always responds to a query $q \in \mathcal{Q}$ with an answer that differs from $q(x)$ by at most α, and every randomized adversary A that adaptively queries M: the view of A when interacting with $M^{\hat{x}}(\cdot,\cdot)$ and the view of A when interacting with $M^{x}(\cdot,\cdot)$ are (ε, δ)-indistinguishable.

Note that this requirement is a generalization of the

differential privacy requirement, since a neighboring database can be viewed as an oracle that provides answers that are close ($\alpha = 1/n$) to those obtained on the true database; that is, *exact* answers on a neighboring database are very close to *exact* answers on the true database (for counting queries, or more generally low-sensitivity queries on x). Furthermore, this also applies to group differential privacy where the goal is to hide whether a group is inside or outside the database (for small groups). The standard Laplace mechanism [2] using independent noise of magnitude $O(\alpha \cdot t/\varepsilon)$ (added to the oracle answer rather than the evaluation of the query on the database) provides $(\varepsilon, 0)$ privacy for α-accurate oracles. (Noise of magnitude $O(\sqrt{t \log(1/\delta)}/(\varepsilon n))$ suffices for (ε, δ) differential privacy.)

However it is not true that any (sufficiently) good differentially private mechanism provides privacy for α-accurate oracles as in Definition III.8. To see this, consider an α-accurate oracle $\hat{x}$ that on query $q \in \mathcal{Q}$ simply takes $q(x)$, the true answer on x, and adds to it at random $+1/n$ or $-1/n$. Given oracle access to $\hat{x}$ it is very easy to distinguish it from one always yielding $q(x)$ simply by checking for consistency. Now take any good differentially private mechanism that is oracle aided, and modify it so that it remembers previous queries to its oracle, and the oracle's responses. If at any point it detects that the same query receives two different values from the oracle, it leaks this fact as follows: It continues with the usual operation of ensuring differential privacy, but encodes in the least significant bit of its outputs whether the oracle is the randomized $\hat{x}$ or simply the one yielding $q(x)$. The differential privacy properties are not affected, but obviously the mechanism does not hide the oracle.

Nevertheless, we observe that the Private Multiplicative Weights algorithm can be used for this purpose and, furthermore, that the analysis of the Private Multiplicative Weights algorithms extends to this more general notion:

Theorem III.9 (Private Multiplicative Weights for α-accurate Oracles)**.** *Let $\mathcal{Q} = \{0,1\}^{\mathcal{U}}$ be the set of all counting queries on data universe $\mathcal{U}$, and let $\alpha_0, \beta, \delta, \varepsilon > 0$ be real numbers, and let $t \in \mathcal{N}$. There is a stateful mechanism that:*

- *Ensures (ε, δ) privacy for α_0-accurate oracles,*
- *Is (α, β) accurate for up to t queries on databases of size n, where α is*

$$O\left(\left(\log^{1/4}|\mathcal{U}|\right)\left(\log^{1/4}(1/\delta)\right)\left(\log^{1/2}(t/\beta)\right)\left(\frac{\alpha_0}{\varepsilon}\right)^{1/2}\right).$$

Note that the major loss is a square root deterioration in accuracy, i.e. α is $\tilde{O}(\sqrt{\alpha_0})$.

Now, to obtain Theorem III.5, we set the parameters in Theorems III.6 and III.9. We take $\mathrm{PMW}_{\mathrm{inner}}$ to be an (ε, δ) differentially private and $(\alpha_0, \beta/6)$ accurate mechanism with

$$\alpha_0 = O\left(\frac{\left(\log^{1/4}|\mathcal{U}|\right)\left(\log^{1/4}(1/\delta)\right)\left(\log^{1/2}(t/\beta)\right)}{(\varepsilon n)^{1/2}}\right).$$

We take each $\mathrm{PMW}_{\mathrm{id}}$ to provide $(\varepsilon'/2, \beta/6)$ privacy for α_0-accurate oracles, and to be $(\alpha, \beta/2t)$ accurate for α equal to

$$O\left(\left(\log^{1/4}|\mathcal{U}|\right)\left(\log^{1/4}(2/\beta)\right)\left(\log^{1/2}(2t^2/\beta)\right)\left(\frac{\alpha_0}{(\varepsilon'/2)}\right)^{1/2}\right)$$
$$=$$
$$O\left(\frac{\left(\log^{3/8}|\mathcal{U}|\right)\left(\log^{1/8}(1/\delta)\right)\left(\log^{1/4}(1/\beta)\right)\left(\log^{3/4}(t/\beta)\right)}{(\varepsilon n)^{1/4} \cdot (\varepsilon')^{1/2}}\right).$$

Accuracy: There is one $\mathrm{PMW}_{\mathrm{inner}}$ executed and since there are at most t queries there are at most t different executions of $\mathrm{PMW}_{\mathrm{id}}$. By a union bound, the probability that accuracy fails for any of the invocations of PMW is at most $\beta/6 + t \cdot (\beta/2t) < \beta$, so we have (α, β) accuracy for the combined mechanism.

Privacy of the analysts: To show privacy for the analyst, fix an id of the adversarial analyst. Whatever queries the other "honest" analysts ask, $\mathrm{PMW}_{\mathrm{inner}}$ still provides an α-accurate oracle to $\mathrm{PMW}_{\mathrm{id}}$, except with probability $\beta/6$. Since $\mathrm{PMW}_{\mathrm{id}}$ is chosen to have $(\varepsilon'/2, \beta/6)$ privacy for α-accurate oracles, the view of the adversarial analyst is $(\varepsilon'/2, \beta/3)$ indistinguishable from its view if we replace $\mathrm{PMW}_{\mathrm{inner}}$ with the actual database x. Thus every two strategies for the "honest" analysts are (ε', β') indistinguishable to the adversary for $\beta' = (1 + e^{\varepsilon'/2}) \cdot (\beta/3) \leq \beta$.

Differential Privacy of the data: There is a single instance of $\mathrm{PMW}_{\mathrm{inner}}$ and the data is only accessed through it. So what an adversary sees is a (randomized) function of the output of $\mathrm{PMW}_{\mathrm{inner}}$. Therefore the differential privacy properties are maintained and we get (ε, δ) differential privacy.

IV. Open Problems

This work opens a new direction for differentially private data analysis: protecting the privacy of the analyst. Many intriguing problems remain.

Collusion: Our Analyst-Private Mechanism from Section III-B only provides many-to-one privacy for the analysts (Definition III.4), meaning that the queries of many analysts are kept private against one adversarial analyst. The analyst privacy does not resist collusion by many adversarial analysts (in contrast to the privacy for the data subjects, which resists even full collusion). In particular, a natural and interesting goal is to achieve one-to-many analyst privacy, where the queries made under any one ID are hidden from all other analysts (even if they collude). An ultimate goal would be to

entirely remove the use of IDs and achieve many-to-many privacy, where any subset of queries is hidden from an adversary controlling all of the remaining queries.

Better Utility: The utility of our analyst-private mechanism (Theorem III.5) does not quite match that of differentially private algorithms that do not provide privacy for the analyst (Theorem III.6). First, as the database size n grows, the error only decays proportionally to $1/n^{1/4}$ instead of $1/n^{1/2}$. Second, the maximum number t of queries that can be answered while providing nontrivial error is $2^{\tilde{\Omega}(n^{1/3})}$ instead of $2^{\tilde{\Omega}(n)}$. Can these gaps be closed or are they an inherent price of providing privacy for the data analyst in addition to the data subjects?

Communication / Query Tradeoff: As noted in the Introduction, our negative result can be interpreted as a negative result about *distributing the work* of answering queries among servers while maintaining differential privacy: either the servers must share information about what queries are asked to them, or they can only answer a small number of queries. Is there a tradeoff between amount of communication and number of queries that can safely be answered with non-trivial accuracy?

Other types of queries: Another issue is for what other types of queries (e.g. low sensitivity queries) do we have mechanisms that preserve the privacy of the analysts. Is there a general method that translates any differentially private mechanism into one that is secure in this sense?

ACKNOWLEDGMENTS

We thank the anonymous reviewers for helpful comments and corrections.

REFERENCES

[1] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)(2)*, 2006, pp. 1–12.

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Theory of Cryptography Conference*, 2006, pp. 265–284.

[3] C. Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, 2011.

[4] "American library association, "other policies and guidelines"," http://www.ala.org/Template.cfm?Section=otherpolicies&Template=/ContentManagement/ContentDisplay.cfm&ContentID=13084, [Online; Accessed April 04, 2012].

[5] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," *JACM*, 1998, preliminary version in FOCS, 1995.

[6] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. N. Wright, "Secure multiparty computation of approximations," in *ICALP*, ser. Lecture Notes in Computer Science, F. Orejas, P. G. Spirakis, and J. van Leeuwen, Eds., vol. 2076. Springer, 2001, pp. 927–938.

[7] M. Naor and V. Teague, "Anti-presistence: history independent data structures," in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*. New York: ACM, 2001, pp. 492–501. [Online]. Available: http://dx.doi.org/10.1145/380752.380844

[8] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 2003, pp. 202–210.

[9] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: privacy via distributed noise generation," in *Advances in Cryptology: Proceedings of EUROCRYPT*, 2006, pp. 486–503.

[10] A. Beimel, K. Nissim, and E. Omri, "Distributed private data analysis: Simultaneously solving how and what," in *CRYPTO*, ser. Lecture Notes in Computer Science, D. Wagner, Ed., vol. 5157. Springer, 2008, pp. 451–468.

[11] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to non-interactive database privacy," in *Proceedings of the 40th ACM SIGACT Symposium on Theory of Computing*, 2008.

[12] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan, "On the complexity of differentially private data release: efficient algorithms and hardness results," in *STOC*, 2009, pp. 381–390.

[13] A. Roth and T. Roughgarden, "The median mechanism: Interactive and efficient privacy with multiple queries," in *Proc. of STOC*, pp. 765–774.

[14] C. Dwork, G. N. Rothblum, and S. P. Vadhan, "Boosting and Differential Privacy," in *Proc. FOCS*, 2010, pp. 51–60.

[15] M. Hardt and G. N. Rothblum, "A multiplicative weights mechanism for interactive privacy-preserving data analysis," *Proc. FOCS*, 2010.

[16] A. Gupta, M. Hardt, A. Roth, and J. Ullman, "Privately releasing conjunctions and the statistical query barrier," in *STOC*, 2011, pp. 803–812.

[17] M. Hardt, K. Ligett, and F. McSherry, "A simple and practical algorithm for differentially private data release," 2010, coRR abs/1012.4763.

[18] M. Bellare, O. Goldreich, and M. Sudan, "Free bits, PCPs, and nonapproximability—towards tight results," *SIAM Journal on Computing*, vol. 27, no. 3, pp. 804–915, 1998. [Online]. Available: http://dx.doi.org/10.1137/S0097539796302531

[19] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and S. Yekhanin, "Pan-private streaming algorithms," in *In Proceedings of ICS*, 2010.

[20] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *STOC*, 2010, pp. 715–724.

[21] S. P. Kasiviswanathan, M. Rudelson, A. Smith, and J. Ullman, "The price of privately releasing contingency tables and the spectra of random matrices with correlated rows," in *STOC*, 2011.

[22] J. Ullman, "Answering n^{2+o(1)} Counting Queries with Differential Privacy is Hard," *ArXiv e-prints*, Jul. 2012.

[23] O. Goldreich and L. A. Levin, "A hard-core predicate for all one-way functions," in *STOC*, 1989, pp. 25–32.

[24] A. Gupta, A. Roth, and J. Ullman, "Iterative constructions and private data release," in *Proceedings of the 9th IACR Theory of Cryptography Conference (TCC '12)*, ser. Lecture Notes on Computer Science. Springer-Verlag, 2012, full version posted as CoRR abs/1107.3731.

[25] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," 2012, monograph in preparation.