Differential Privacy for the Analyst via Private Equilibrium Computation[*]

Justin Hsu[†]
University of Pennsylvania
justhsu@cis.upenn.edu

Aaron Roth[‡]
University of Pennsylvania
aaroth@cis.upenn.edu

Jonathan Ullman[§]
Harvard University
jullman@seas.harvard.edu

ABSTRACT

We give new mechanisms for answering exponentially many queries from multiple analysts on a private database, while protecting differential privacy both for the individuals in the database and for the analysts. That is, our mechanism's answer to each query is nearly insensitive to changes in the queries asked by other analysts. Our mechanism is the first to offer differential privacy on the joint distribution over analysts' answers, providing privacy for data analysts even if the other data analysts collude or register multiple accounts. In some settings, we are able to achieve nearly optimal error rates (even compared to mechanisms which do not offer analyst privacy), and we are able to extend our techniques to handle non-linear queries. Our analysis is based on a novel view of the private query-release problem as a two-player zero-sum game, which may be of independent interest.

Categories and Subject Descriptors

F.0 [**THEORY OF COMPUTATION**]: General

Keywords

Differential Privacy; Analyst Privacy; No-regret Learning; Bregman Projection; Zero-sum Game

1. INTRODUCTION

Consider a tracking network that wants to sell a database of consumer data to several competing analysts conducting market research. The administrator of the tracking network faces many opposing constraints when deciding how to provide analysts with this data. For legal reasons, the privacy of the individuals contained in her database must be protected. At the same time, the analysts must be able to query the database and receive useful answers. Finally, the privacy of the *queries* made to the database must be protected, since the analysts are in competition and their queries may be disclosive of proprietary strategies.

This setting of *analyst privacy* was recently introduced in a beautiful paper of Dwork, Naor, and Vadhan [7]. They showed that differentially private *stateless* mechanisms — which answer each query *independently* of the previous queries — can only give accurate answers when the number of queries is at most quadratic in the size of the database. This result rules out mechanisms that perfectly protect the privacy of the queries, while accurately answering exponentially many queries — answers must depend on the state, and hence on the previous queries. However, it turns out that mechanisms that offer a differential-privacy-like guarantee with respect to the queries are possible: Dwork, et al. [7] give such a mechanism, with the guarantee that the marginal distribution on answers given to each analyst is differentially private with respect to the set of queries made by all of the other analysts. Their mechanism is capable of answering exponentially many linear queries with error $\widetilde{O}(1/n^{1/4})$, where n is the number of records in the database. A *linear query* is a $(1/n)$-sensitive query of the form "What fraction of the individual records in the database satisfy some property q?", so their mechanism gives non-trivial accuracy.

However, they note that their mechanism has several shortcomings. First, it does not promise differential privacy on the *joint distribution* over multiple analysts' answers. Therefore, if multiple analysts collude, or if a single malicious analyst registers several false accounts with the mechanism, then the mechanism no longer guarantees query privacy. Second, their mechanism is less accurate than known, non-analyst private mechanisms — analyst privacy is achieved at a cost to accuracy. Finally, the mechanism can only answer linear queries, rather than general low-sensitivity queries.

In this paper, we address all of these issues. First, we consider mechanisms which guarantee *one-query-to-many-analyst* privacy: for each analyst a, the joint distribution over answers given to *all other* analysts $a' \neq a$ is differentially private with respect to the change of a single query asked by analyst a. This privacy guarantee is incomparable to that of Dwork, et al. [7]: it is weaker, because we protect the privacy of a single query, rather than protecting the privacy of all queries asked by analysts $a' \neq a$. However, it is also stronger, because the privacy of one query from an analyst a is preserved even if all other analysts $a' \neq a$ collude or register multiple accounts. Our first result is a mechanism in this setting, with error at most $\widetilde{O}(1/\sqrt{n})$ for answering exponentially many linear queries. This error is optimal up to polylogarithmic factors, even when comparing to mechanisms that only guarantee data privacy.

We then extend our techniques to *one-analyst-to-many-analyst* privacy, where we require that the mechanism preserve the privacy of analyst when he changes *all* of his queries, even if *all other* an-

[*] A full version of this work appears on arXiv [14].

[†] Supported in part by NSF grant CNS-1065060.

[‡] Supported in part by NSF grant CCF-1101389 and CNS-1065060.

[§] Harvard University School of Engineering and Applied Sciences. Supported by NSF grant CNS-1237235 and a Siebel Scholarship.

STOC'13, June 1-4, 2013, Palo Alto, California, USA.
Copyright 2013 ACM 978-1-4503-2029-0/13/06 ...$15.00.

alysts collude. Our second result is a mechanism in this setting, with error $\widetilde{O}(1/n^{1/3})$. Although this error rate is worse than what we achieve for one-query-to-many-analyst privacy (and not necessarily optimal), our mechanism is still capable of answering exponentially many queries with non-trivial accuracy guarantees, while satisfying both data and analyst privacy.

These first two mechanisms operate in the *non-interactive* setting, where the queries from every analyst are given to the mechanism in a single batch. Our final result is a mechanism in the *online* setting that satisfies one-query-to-many analyst privacy. The mechanism accurately answers a (possibly exponentially long) *fixed* sequence of low-sensitivity queries. Although our mechanism operates as queries arrive online, it cannot tolerate adversarially chosen queries (i.e. it operates in the same regime as the *smooth multiplicative weights* algorithm of Hardt and Rothblum [13]). For linear queries, our mechanism gives answers with error at most $\widetilde{O}(1/n^{2/5})$. For answering general queries with sensitivity $1/n$ (the sensitivity of a linear query), the mechanism guarantees error at most $\widetilde{O}(1/n^{1/10})$.

When answering k queries on a database $D \in \mathcal{X}^n$ consisting of n records from a data universe $\mathcal{X}$, our offline algorithms run in time $\widetilde{O}(n \cdot (|\mathcal{X}| + k))$ and our online algorithm runs in time $\widetilde{O}(|\mathcal{X}| + n)$ per query. These running times are essentially optimal for mechanisms that answer more than $\omega(n^2)$ arbitrary linear queries [22], assuming (exponentially hard) one-way functions exist.

Our Techniques. To prove our results, we take a novel view of private query release as a two player zero-sum game between a *data player* and a *query player*. For each element of the data universe $x \in X$, the data player has an action a_x. Intuitively, the data player's mixed strategy will be his approximation of the true database's distribution.

On the other side, for each query $q \in \mathcal{Q}$, the query player has two actions: a_q and $a_{\neg q}$. The two actions for each query allow the query player to penalize the data player's play, both when the approximate answer to q is too high, and when it is too low — the query player tries to play queries for which the data player's approximation performs poorly. Formally, we define the cost matrix by $G(a_q, a_x) = q(x) - q(D)$ and $G(a_{\neg q}, a_x) = q(D) - q(x)$, where D is the private database. The query player wishes to maximize the cost, whereas the database player wishes to minimize the cost. We show that the value of this game is 0, and that any ρ-approximate equilibrium strategy for the database player corresponds to a database that answers every query $q \in \mathcal{Q}$ correctly up to additive error ρ. Thus, given any pair of ρ-approximate equilibrium strategies, the strategy for the data player will constitute a database (a distribution over $\mathcal{X}$) that answers every query to within error $O(\rho)$.

Different privacy constraints for the private query release problem can be mapped into privacy constraints for solving two-player zero-sum games. Standard private linear query release corresponds to privately computing an approximate equilibrium, where privacy is preserved with respect to changing every cost in the game matrix by at most $1/n$. Likewise, query release while protecting *one-query-to-many-analyst* privacy corresponds to computing an approximate equilibrium strategy, where privacy is with respect to an arbitrary change in two *rows* of the game matrix — changing a single query q changes the payoffs for actions a_q and $a_{\neg q}$. Our main result can be viewed as an algorithm for privately computing the equilibrium of a zero-sum game while protecting the privacy of strategies of the players, which may be of independent interest.

To construct an approximate equilibrium, we use a well-known result: when two no-regret algorithms are played against each other in a zero-sum game, their empirical play distributions quickly converge to an approximate equilibrium. Thus, to compute an equilibrium of G, we have the query player and the data player play against each other using no-regret algorithms, and output the empirical play distribution of the data player as the hypothesis database. We face several obstacles along the way.

First, no-regret algorithms maintain a *state* — a distribution over actions, which is not privacy preserving. (In fact, it is computed deterministically from inputs that may depend on the data or queries.) Previous approaches to private query release have addressed this problem by adding noise to the *inputs* of the no-regret algorithm.

In our approach, we crucially rely on the fact that *sampling* actions from the distributions maintained by the multiplicative weights algorithm is privacy preserving. Intuitively, privacy will come from the fact that the multiplicative weights algorithm does not adjust the weight on any action too aggressively, meaning that when we view the weights as defining a distribution over actions, changing the losses experienced by the algorithm in various ways will have a limited effect on the distribution over actions. We note that this property is not used in the private multiplicative weights mechanism of Hardt and Rothblum [13], who use the distribution itself as a hypothesis. Indeed, without the constraint of query privacy, *any* no-regret algorithm can be used in place of multiplicative weights [21, 11], which is not the case in our setting.

Second, sampling from the multiplicative weights algorithm is private only if the changes in losses are small. Intuitively, we must ensure that changing one query from one analyst does not affect the losses experienced by the data player too dramatically, so that samples from the multiplicative weights algorithm will indeed ensure query privacy. To enforce this requirement, we force the query player to play mixed strategies from the set of *smooth* distributions, which do not place too much weight on any single action. It is known that playing any no-regret algorithm, but projecting into the set of smooth distributions in the appropriate way (via a Bregman projection), will ensure no-regret with respect to any smooth distribution on actions. For comparison, no-regret is typically defined with respect to the best single action, which is a not a smooth distribution. Thus, our regret guarantee is weaker.

The result of this simulation is an approximate equilibrium strategy for the data player, in the sense that it achieves approximately the value of the game when played against *all but s* strategies of the query player, where $1/s$ is the maximum probability that the query player may assign to any action. This corresponds to a synthetic database, which we release to all analysts, that answers *all but s* queries accurately. Then, since we choose s to be small, we can answer the mishandled queries with the sparse vector technique [6, 21, 13] adding noise only $\widetilde{O}(\sqrt{s}/n)$ to these s queries. The result is a nearly optimal error rate of $\widetilde{O}(1/\sqrt{n})$

Our techniques naturally extend to one-analyst-to-many-analyst privacy by making the actions of the query player correspond to entire workloads of queries, one for each analyst, where the query player picks analysts that have at least one query that has high error on the current hypothesis. Like before, a small number of analysts will have queries that have high error, which we handle with a separate private query release mechanism for each analyst.

Finally, we use these techniques to convert the private multiplicative weights algorithm of Hardt and Rothblum [13] into an online algorithm that preserves one-query-to-many-analyst privacy, and also answers arbitrary low-sensitivity queries. These last two extensions both give first-of-their-kind results, but at some degradation in the accuracy parameters: we do not obtain $O(1/\sqrt{n})$ error rate. We leave it as an open problem to achieve $\widetilde{O}(1/\sqrt{n})$ error in these settings, or show that the accuracy cost is necessary.

Related Work. There is an extremely large body of work on differential privacy [5] that we do not attempt to survey. The study of differential privacy was initiated by a line of work [4, 2, 5] culminating in the definition by Dwork, Mcsherry, Nissim, and Smith [5], who also introduced the basic technique of answering low-sensitivity queries using the Laplace mechanism. The Laplace mechanism gives approximate answers to nearly $O(n^2)$ queries while preserving differential privacy.

A recent line of work [3, 6, 8, 21, 13, 10, 11, 12] has shown how to accurately answer almost *exponentially many* queries usefully while preserving differential privacy of the data. Some of this work [21, 13, 11] rely on *no-regret* algorithms — in particular, Hardt and Rothblum [13] introduced the multiplicative weights technique to the differential privacy literature, which we use centrally.

However, we make use of multiplicative weights in a different way from prior work in private query release — we simulate play of a two-player zero-sum game using *two* copies of the multiplicative weights algorithm, and rely on the fast convergence of such play to approximate Nash equilibrium [9]. We also rely on the fact that Bregman projections onto a convex set K can be used in conjunction with the multiplicative weights update rule to achieve no regret with respect to the best element in the set K [19]. Finally, we use that *samples* from the multiplicative weights distribution can be viewed as samples from the exponential mechanism of McSherry and Talwar [18], and hence are privacy preserving.

Our use of Bregman projections into smooth distributions is similar to its use in *smooth boosting*. Barak, Hardt, and Kale [1] use Bregman projections in a similar way, and the weight capping used by Dwork, Rothblum, and Vadhan [8] in their analysis of *boosting for people* can be viewed as a Bregman projection.

The most closely related paper to ours is the beautiful recent work of Dwork, Naor, and Vadhan [7], who introduce the idea of analyst privacy. They show that any algorithm which can answer $\omega(n^2)$ queries to non-trivial accuracy must maintain *common state* as it interacts with many data analysts, and hence potentially violates the privacy of the analysts. Accordingly, they give a stateful mechanism which promises many-to-one-analyst privacy, and achieves per-query error $\widetilde{O}(1/n^{1/4})$ for linear queries — their mechanism promises differential privacy on the marginal distribution of answers given to any single analyst, even when all other analysts change all of their queries. However, if multiple analysts collude, or if a single analyst can falsely register under many ids, then the privacy guarantees degrade quickly — privacy is not promised on the *joint distribution* on all analysts answers. Lifting this limitation, improving the error bounds and extending analyst privacy to non-linear queries, are all stated as open questions.

Finally, the varying notions of analyst privacy we use can be interpreted in the context of two-party differential privacy, introduced by McGregor, et al. [17]. If we consider a single analyst as one party, sending private queries to a second party consisting of the mechanism and all the other parties indirectly, the many-to-one-analyst privacy guarantee is equivalent to privacy of the first party's view. Here, the privacy must be protected even if the second party changes its inputs arbitrarily, i.e., the other analysts change their queries arbitrarily.

On other hand, if we consider all but one analyst as the first party, sending queries to the mechanism and the remaining analyst, one-query-to-many-analyst privacy is equivalent to the first party's view being private when the single analyst changes a query. One-analyst-to-many-analyst privacy is similar: the first party's view must be private when the second party changes all of its queries.

2. PRELIMINARIES

Differential Privacy and Analyst Differential Privacy. Let a *database* $D \in \mathcal{X}^n$ be a collection of n records (rows) $\{x^{(1)}, \ldots, x^{(n)}\}$ from a *data universe* $\mathcal{X}$. Two databases $D, D' \in \mathcal{X}^n$ are *adjacent* if they differ only on a single row, which we denote by $D \sim D'$.

A mechanism $\mathcal{A} : \mathcal{X}^n \to \mathcal{R}$ takes a database as input and outputs some data structure in $\mathcal{R}$. We are interested in mechanisms that satisfy *differential privacy*.

DEFINITION 2.1. *A mechanism $\mathcal{A} : \mathcal{X}^n \to \mathcal{R}$ is (ε, δ)*-differentially private *if for every two adjacent databases $D \sim D' \in \mathcal{X}^n$ and every subset $S \subseteq \mathcal{R}$,*

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{A}(D') \in S] + \delta.$$

In this work we construct mechanisms that ensure differential privacy *for the analyst* as well as for the database. To define analyst privacy, we first define *many-analyst mechanisms*. Let $\mathbf{Q}$ be the set of all allowable queries. The mechanism takes m sets of queries $\mathcal{Q}_1, \ldots, \mathcal{Q}_m$ and returns m outputs $Z_1, \ldots, Z_m$, where Z_j contains answers to the queries $\mathcal{Q}_j$. Thus, a many-analyst mechanism has the form $\mathcal{A} : \mathcal{X}^n \times (\mathbf{Q}^*)^m \to \mathcal{R}^m$. Given sets of queries $\mathcal{Q}_1, \ldots, \mathcal{Q}_m$, let $\mathcal{Q} = \bigcup_{j=1}^m \mathcal{Q}_j$ denote the set of all queries. In guaranteeing privacy even in the event of collusion, it will be useful to refer to the output given to all analysts other than some analyst i. For each id $\in [m]$ we write $\mathcal{A}(D, \mathcal{Q})_{-\mathrm{id}}$ to denote $(Z_1, \ldots, Z_{\mathrm{id}-1}, Z_{\mathrm{id}+1}, \ldots, Z_m)$, the output given to all analysts other than id.

Let $\mathcal{Q} = \mathcal{Q}_1, \ldots, \mathcal{Q}_m$ and $\mathcal{Q}' = \mathcal{Q}'_1, \ldots, \mathcal{Q}'_m$. We say that $\mathcal{Q}$ and $\mathcal{Q}'$ are *analyst-adjacent* if there exists id$^* \in [m]$ such that for every id $\neq$ id*, $\mathcal{Q}_{\mathrm{id}} = \mathcal{Q}'_{\mathrm{id}}$. That is, $\mathcal{Q} \sim \mathcal{Q}'$ are analyst adjacent if they differ only on the queries asked by one analyst. Intuitively, a mechanism satisfies one-analyst-to-many-analyst privacy if changing *all the queries asked by analyst* id* does not significantly affect the output given to *all analysts other than* id*.

DEFINITION 2.2. *A many-analyst mechanism $\mathcal{A}$ satisfies (ε, δ)-*one-analyst-to-many-analyst privacy *if for every database $D \in \mathcal{X}^n$, every two analyst-adjacent query sequences $\mathcal{Q} \sim \mathcal{Q}'$ that differ only on one set of queries $\mathcal{Q}_{\mathrm{id}}, \mathcal{Q}'_{\mathrm{id}}$, and every $S \subseteq \mathcal{R}^{m-1}$,*

$$\Pr[\mathcal{A}(D, \mathcal{Q})_{-\mathrm{id}} \in S] \leq e^{\varepsilon} \Pr[\mathcal{A}(D, \mathcal{Q}')_{-\mathrm{id}} \in S] + \delta.$$

Let $\mathcal{Q} = \mathcal{Q}_1, \ldots, \mathcal{Q}_m$ and $\mathcal{Q}' = \mathcal{Q}'_1, \ldots, \mathcal{Q}'_m$. We say that $\mathcal{Q}$ and $\mathcal{Q}'$ are *query-adjacent* if there exists id* such that for every id $\neq$ id*, $\mathcal{Q}_{\mathrm{id}} = \mathcal{Q}'_{\mathrm{id}}$ and $|\mathcal{Q}_{\mathrm{id}^*} \triangle \mathcal{Q}'_{\mathrm{id}^*}| \leq 1$. That is, $\mathcal{Q} \sim \mathcal{Q}'$ are query adjacent if they differ only on one of the queries. Intuitively, we say that a mechanism satisfies one-query-to-many-analyst privacy if changing *one query asked by analyst* id* does not significantly affect the output given to *all analysts other than* id*.

DEFINITION 2.3. *A many-analyst mechanism $\mathcal{A}$ satisfies (ε, δ)-*one-query-to-many-analyst privacy *if for every database $D \in \mathcal{X}^n$, every two query-adjacent query sequences $\mathcal{Q} \sim \mathcal{Q}'$ that differ only on one query in $\mathcal{Q}_{\mathrm{id}}, \mathcal{Q}'_{\mathrm{id}}$, and every $S \subseteq \mathcal{R}^{m-1}$,*

$$\Pr[\mathcal{A}(D, \mathcal{Q})_{-\mathrm{id}} \in S] \leq e^{\varepsilon} \Pr[\mathcal{A}(D, \mathcal{Q}')_{-\mathrm{id}} \in S] + \delta.$$

In our proofs of both differential privacy and analyst privacy, we will often establish that for any $D \sim D'$, the two distributions

$\mathcal{A}(D), \mathcal{A}(D')$ are such that with probability at least $1-\delta$ over $y \leftarrow_{\text{R}} M(D)$,

$$\left|\ln\left(\frac{\Pr[\mathcal{A}(D)=y]}{\Pr[\mathcal{A}(D')=y]}\right)\right| \le \varepsilon.$$

This condition implies (ε, δ)-differential privacy [8].

Queries and Accuracy. In this work we consider two types of queries: *low-sensitivity queries* and *linear queries*. Low-sensitivity queries are parameterized by $\Delta \in [0,1]$: a Δ-*sensitive query* is any function $q : \mathcal{X}^n \to [0,1]$ such that $\max_{D \sim D'} |q(D) - q(D')| \le \Delta$. A *linear query* is a particular type of low-sensitivity query, specified by a function $q : \mathcal{X} \to [0,1]$. We define the evaluation of q on a database $D \in \mathcal{X}^n$ to be $q(D) = \frac{1}{n}\sum_{i=1}^{n} q(x^{(i)})$, so a linear query is evidently $(1/n)$-sensitive.

Since $\mathcal{A}$ may output a data structure, we must specify how to answer queries in $\mathcal{Q}$ from the output $\mathcal{A}(D)$. Hence, we require that there is an *evaluator* $\mathcal{E} : \mathcal{R} \times \mathcal{Q} \to \mathbb{R}$ that estimates $q(D)$ from the output of $\mathcal{A}(D)$. For example, if $\mathcal{A}$ outputs a vector of "noisy answers" $Z = \{q(D) + Z_q | q \in \mathcal{Q}\}$, where Z_q is a random variable for each $q \in \mathcal{Q}$, then $\mathcal{R} = \mathbb{R}^{\mathcal{Q}}$ and $\mathcal{E}(Z, q)$ is the q-th component of Z. Abusing notation, we write $q(Z)$ and $q(\mathcal{A}(D))$ as shorthand for $\mathcal{E}(Z, q)$ and $\mathcal{E}(\mathcal{A}(D), q)$, respectively.

DEFINITION 2.4. *An output Z of a mechanism $\mathcal{A}(D)$ is α-*accurate *for query set $\mathcal{Q}$ if $|q(Z) - q(D)| \le \alpha$ for every $q \in \mathcal{Q}$. A mechanism is (α, β)-*accurate *for query set $\mathcal{Q}$ if for every database D,*

$$\Pr\left[\forall q \in \mathcal{Q}, |q(\mathcal{A}(D)) - q(D)| \le \alpha\right] \ge 1 - \beta,$$

where the probability is taken over the coins of $\mathcal{A}$.

Differential Privacy Tools. We will use a few previously known differentially private mechanisms. When we need to answer a small number of queries we will use the well-known Laplace mechanism [5], with an improved analysis from [8].

LEMMA 2.5. *Let $\mathcal{F} = \left\{f_1, \ldots, f_{|\mathcal{F}|}\right\}$ be a set of Δ-sensitive queries $f_i : \mathcal{X}^n \to [0,1]$, and let $D \in \mathcal{X}^n$ be a database. Let $\epsilon, \delta \le 1$. Then the mechanism $\mathcal{A}_{\text{Lap}}(D, \mathcal{F})$ that outputs $f_i(D) + \text{Lap}\left(\frac{\Delta\sqrt{8|\mathcal{F}|\log(1/\delta)}}{\varepsilon}\right)$ for every $f_i \in \mathcal{F}$ is:*

1. *(ε, δ)-differentially private, and*
2. *(α, β)-accurate for any $\beta \in (0, 1]$ and $\alpha = \varepsilon^{-1}\Delta\sqrt{8|\mathcal{F}|\log(1/\delta)}\log(|\mathcal{F}|/\beta)$.*

When we need to answer a large number of queries, we will use the multiplicative weights mechanism from [13], with an improved analysis from Gupta et al. [11].

LEMMA 2.6. *Let $\mathcal{F} = \left\{f_1, \ldots, f_{|\mathcal{F}|}\right\}$ be a set of $(1/n)$-sensitive linear queries, $f_i : \mathcal{X}^n \to [0,1]$. Let $D \in \mathcal{X}^n$ be a database. Then there is a mechanism $\mathcal{A}_{\text{MW}}(D, \mathcal{F})$ that is:*

1. *(ε, δ)-differentially private, and*
2. *(α, β)-accurate for any $\beta \in (0, 1]$ and $\alpha = O\left(\frac{\log^{1/4}|\mathcal{X}|\sqrt{\log(|\mathcal{F}|/\beta)\log(1/\delta)}}{\varepsilon^{1/2}n^{1/2}}\right)$.*

REMARK 2.7. *We use the above lemma as a black box, agnostic to the algorithm which instantiates these guarantees.*

Our algorithms also use the private sparse vector algorithm. This algorithm takes as input a database and a set of low-sensitivity queries, with the promise that only a small number of the queries have large answers on the input database. Its output is a set of queries with large answers on the input database. Importantly for this work, the sparse vector algorithm (cf. [13, 20]) ensures the privacy of the input queries in a strong sense.

LEMMA 2.8. *Let $\mathcal{F} = \left\{f_1, \ldots, f_{|\mathcal{F}|}\right\}$ be a set of Δ-sensitive functions, $f_i : \mathcal{X}^n \to [0,1]$. Let $D \in \mathcal{X}^n$ be a database, $\alpha \in (0,1]$, $k \in [|\mathcal{F}|]$ such that $|\{i \mid f_i(D) \ge \alpha\}| \le k$. Then there is an algorithm $\mathcal{A}_{\text{SV}}(D, \mathcal{F})$ that*

1. *is (ε, δ)-differentially private with respect to D,*
2. *returns $I \subseteq [|\mathcal{F}|]$ of size at most k such that with probability at least $1 - \beta$,*

$$\left\{i \mid f_i(D) \ge \alpha + \varepsilon^{-1}\Delta\sqrt{8k\log(1/\delta)}\log(|\mathcal{F}|/\beta)\right\} \subseteq I \subseteq \{i \mid f_i(D) \ge \alpha\},$$

3. *and is perfectly private with respect to the queries: if $\mathcal{F}' = \left\{f_1, \ldots, f_j', \ldots, f_k\right\}$, then for every D and $i \ne j$,*

$$\Pr\left[i \in \mathcal{A}_{\text{SV}}(D, \mathcal{F})\right] = \Pr\left[i \in \mathcal{A}_{\text{SV}}(D, \mathcal{F}')\right].$$

We will also use the Composition Theorem of Dwork, Rothblum, and Vadhan [8].

LEMMA 2.9. *Let $\mathcal{A} : \mathcal{X}^* \to \mathcal{R}^T$ be a mechanism such that for every pair of adjacent inputs $x \sim x'$, every $t \in [T]$, every $r_1, \ldots, r_{t-1} \in \mathcal{R}$, and every $r_t \in \mathcal{R}$,*

$$\begin{aligned} &\Pr\left[\mathcal{A}_t(x) = r_t \mid \mathcal{A}_{1,\ldots,t-1}(x) = r_1, \ldots, r_{t-1}\right] \\ &\le e^{\varepsilon_0}\Pr\left[\mathcal{A}_t(x') = r_t \mid \mathcal{A}_{1,\ldots,t-1}(x') = r_1, \ldots, r_{t-1}\right] + \delta_0 \end{aligned}$$

for $\varepsilon_0 \le 1/2$. Then $\mathcal{A}$ is (ε, δ)-differentially private for $\varepsilon = \sqrt{8T\log(1/\delta)} + 2\varepsilon_0^2 T$ and $\delta = \delta_0 T$.

Multiplicative Weights. Let $A : \mathcal{A} \to [0,1]$ be a measure over a set of actions $\mathcal{A}$. We use $|A| = \sum_{a \in \mathcal{A}} A(a)$ to denote the *density* of A. A measure naturally corresponds to a probability distribution $\widetilde{A}$ in which $\Pr\left[\widetilde{A} = a\right] = A(a)/|A|$ for every $a \in \mathcal{A}$. Throughout, we will use calligraphic letters ($\mathcal{A}$) to denote a set of actions, lower case letters (a) to denote the actions, capital letters (A) to denote a measure over actions, and capital letters with a tilde to denote the corresponding distributions ($\widetilde{A}$). Let $L : \mathcal{A} \to [0,1]$ be a loss function (losses L). Abusing notation, we can define $L(A) = \mathbb{E}\left[L(\widetilde{A})\right]$. Given an initial measure A_1, we can define the multiplicative weights algorithm in Algorithm 1.

Algorithm 1 The Multiplicative Weights Algorithm, MW_η

```
For $t = 1, 2, \ldots, T$:
    Sample $a_t \leftarrow_{\text{R}} \widetilde{A}_t$
    Receive losses $L_t$ (may depend on $A_1, a_1, \ldots, A_{t-1}, a_{t-1}$)
    **Update: For each** $a \in \mathcal{A}$**:**
        Update $A_{t+1}(a) = e^{-\eta L_t(a)} A_t(a)$ for every $a \in \mathcal{A}$
```

The following theorem about the multiplicative weights update is well-known.

THEOREM 2.10 (SEE E.G. [19]). *Let A_1 be the uniform measure of density* 1*, and let $\{a_1, \dots, a_T\}$ be the actions obtained by MW_η with losses $\{L_1, \dots, L_t\}$. Let $A^* = \mathbf{1}_{a=a^*}$, for some $a^* \in \mathcal{A}$, and $\delta \in (0, 1]$. Then with probability at least $1 - \beta$,*

$$\mathop{\mathbb{E}}_{t \leftarrow_R [T]} [L_t(a_t)] \leq \mathop{\mathbb{E}}_{t \leftarrow_R [T]} [L_t(A^*)] + \eta + \frac{\log |\mathcal{A}|}{\eta T} + \frac{4 \log(1/\beta)}{\sqrt{T}}.$$

We need to work with a variant of multiplicative weights that only produces measures A of high density, which will imply that $\widetilde{A}$ does not assign too much probability to any single element of $\mathcal{A}$. To this end, we will apply (a special case of) the Bregman projection to the measures obtained from the multiplicative weights update rule.

DEFINITION 2.11. *Let $s \in (0, \mathcal{U}]$. Given a measure A such that $|A| \leq s$, let $\Gamma_s A$ be the* (Bregman) projection of A into the set of density-s measures*, obtained by computing $c \geq 1$ such that $s = \sum_{a \in \mathcal{A}} \min\{1, cA(a)\}$ and setting $\Gamma A(a) = \min\{1, cM(a)\}$ for every $a \in \mathcal{A}$. We call s is the* density *of measure A.*

Algorithm 2 The Dense Multiplicative Weights Algorithm, $DMW_{s,\eta}$

```
For $t = 1, 2, \dots, T$:
    Let $A'_t = \Gamma_s A_t$, and sample $a_t \leftarrow_R \widetilde{A}'_t$
    Receive losses $L_t$ (may depend on $A_1, a_1, \dots, A_{t-1}, a_{t-1}$)
    **Update: For each $a \in \mathcal{A}$:**
        Update $A_{t+1}(a) = e^{-\eta L_t(a)} A_t(a)$
```

Given an initial measure A_1 such that $|A_1| \leq s$, we can define the dense multiplicative weights algorithm in Algorithm 2. Note that we update the unprojected measure A_t, but sample a_t using the projected measure $\Gamma_s A_t$. Observe that the update step can only decrease the density, so we will have $|A_t| \leq s$ for every t. Like before, given a sequence of losses $\{L_1, \dots, L_T\}$ and an initial measure A_1 of density s, we can consider the sequence $\{A_1, \dots, A_T\}$ where A_{t+1} is given by the projected multiplicative weights update applied to A_t, L_t. The following theorem is known.

THEOREM 2.12. *Let A_1 be the uniform measure of density* 1 *and let $\{a_1, \dots, a_T\}$ be the sequence of measures obtained by $DMW_{s,\eta}$ with losses $\{L_1, \dots, L_T\}$. Let $A^* = \mathbf{1}_{a \in S*}$ for some set $S^* \subseteq \mathcal{A}$ of size s, and $\delta \in (0, 1]$. Then with probability $1 - \beta$,*

$$\mathop{\mathbb{E}}_{t \leftarrow_R [T]} [L_t(\Gamma A_t)] \leq \mathop{\mathbb{E}}_{t \leftarrow_R [T]} [L_t(A^*)] + \eta + \frac{\log |\mathcal{A}|}{\eta T} + \frac{4 \log(1/\beta)}{\sqrt{T}}.$$

See e.g. [19] for a thorough treatment of this result.

Regret Minimization and Two-Player Zero-Sum Games.

Let $G : \mathcal{A}_R \times \mathcal{A}_C \to [0, 1]$ be a two-player zero-sum game between players (R)ow and (C)olumn, who take actions $r \in \mathcal{A}_R$ and $c \in \mathcal{A}_C$ and receive losses $G(r, c)$ and $-G(r, c)$, respectively. Let $\Delta(\mathcal{A}_R), \Delta(\mathcal{A}_C)$ be the set of measures over actions in $\mathcal{A}_R$ and $\mathcal{A}_C$, respectively. The well-known minimax theorem states that

$$v := \min_{R \in \Delta(\mathcal{A}_R)} \max_{C \in \Delta(\mathcal{A}_C)} G(R, C) = \max_{C \in \Delta(\mathcal{A}_C)} \min_{R \in \Delta(\mathcal{A}_R)} G(R, C).$$

We define this quantity v to be the *value of the game*.

Freund and Schapire [9] showed that if two sequences of actions $\{r_1, \dots, r_T\}, \{c_1, \dots, c_T\}$ are "no-regret with respect to one another", then $\widetilde{r} = \frac{1}{T} \sum_{t=1}^T r_t$ and $\widetilde{c} = \frac{1}{T} \sum_{t=1}^T c_t$ form an approximate equilibrium strategy pair. More formally, if

$$\max_{c \in \mathcal{A}_C} \mathop{\mathbb{E}}_t [G(r_t, c)] - \rho \leq \mathop{\mathbb{E}}_t [G(r_t, c_t)] \leq \min_{r \in \mathcal{A}_R} \mathop{\mathbb{E}}_t [G(r, c_t)] + \rho,$$

then $v - 2\rho \leq G(\widetilde{r}, \widetilde{c}) \leq v + 2\rho$. Thus, if Row chooses actions using the multiplicative weights update rule with losses $L_t(r_t) = G(r_t, c_t)$ and Column chooses actions using the multiplicative weights rule with losses $L_t(r_t) = -G(r_t, c_t)$, then each player's distribution on actions converges to a minimax strategy.

For query privacy in our view of query release as a two player game, Column must not put too much weight on any single query. Thus, we need an analogue of this result in the case where Column is not choosing actions according to the multiplicative weights update, but rather using the projected multiplicative weights update. In this case we cannot hope to obtain an approximate minimax strategy, since Column cannot play any single action with significant probability. However, we can define an alternative notion of the value of a game where Column is restricted in this way: let $\Delta_s(\mathcal{A}_C)$ be the set of measures over $\mathcal{A}_C$ of minimum density at least s, and define

$$v_s := \min_{R \in \Delta(\mathcal{A}_R)} \max_{C \in \Delta_s(\mathcal{A}_C)} G(R, C).$$

Notice that $v_s \leq v$, and v_s can be very different from v.

THEOREM 2.13. *Let $\{r_1, \dots, r_T\} \in \mathcal{A}_R$ be a sequence of row-player actions, $\{C_1, \dots, C_T\} \in \Delta_s(\mathcal{A}_C)$ be a sequence of high-density measures over column-player actions, and $\{c_1, \dots, c_T\} \in \mathcal{A}_C$ be a sequence of column-player actions such that $c_j \leftarrow_R C_j$ for every $t \in [T]$. Further, suppose that*

$$\max_{C \in \Delta_s(\mathcal{A}_C)} \mathop{\mathbb{E}}_t [G(r_t, C)] - \rho$$
$$\leq \mathop{\mathbb{E}}_t [G(r_t, c_t)] \leq \min_{R \in \Delta(\mathcal{A}_R)} \mathop{\mathbb{E}}_t [G(R, c_t)] + \rho$$

Then, $v_s - 2\rho \leq G(\widetilde{r}, \widetilde{c}) \leq v + 2\rho$. Moreover, $\widetilde{r}$ is an approximate min-max strategy with respect to strategies in $\Delta_s(\mathcal{A}_C)$, i.e., $v_s - 2\rho \leq \max_{C \in \Delta_s(\mathcal{A}_C)} G(\widetilde{r}, C) \leq v + 2\rho$.

We omit the proof, which closely follows the argument of Freund and Schapire [9] for the unconstrained case.

COROLLARY 2.14. *Let $G : \mathcal{A}_R \times \mathcal{A}_C \to [0, 1]$. If the row player chooses actions $\{r_1, \dots, r_T\}$ by running MW_η with loss functions $L_t(r) = G(r, c_t)$ and the column player chooses actions $\{c_1, \dots, c_T\}$ by running $DMW_{s,\eta}$ with the loss functions $L_t(c) = -G(r_t, c)$, then with probability at least $1 - \beta$, $v_s - 2\rho \leq \max_{c \in C_s} G(\widetilde{r}, c) \leq v + 2\rho$, for*

$$\rho = \eta + \frac{\max\{\log |\mathcal{A}_R|, \log |\mathcal{A}_C|\}}{\eta T} + \frac{4 \log(2/\beta)}{\sqrt{T}}.$$

3. A ONE-QUERY-TO-MANY-ANALYST PRIVATE MECHANISM

We define our offline mechanisms for releasing linear queries in Algorithm 3.

Accuracy Analysis

THEOREM 3.1. *The offline algorithm for linear queries is (α, β)-accurate for*

$$\alpha = O\left(\frac{\sqrt{\log(|\mathcal{X}| + |\mathcal{Q}|) \log(1/\delta)} \log(|\mathcal{Q}|/\beta)}{\varepsilon \sqrt{n}}\right).$$

Algorithm 3 Offline Mechanism for Linear Queries with One-Query-to-Many-Analyst Privacy

```
Input: Database $D \in \mathcal{X}^n$ and sets of linear queries $\mathcal{Q}_1, \ldots, \mathcal{Q}_m$.
Initialize: Let $\mathcal{Q} = \bigcup_{j=1}^m \mathcal{Q}_j \cup \neg\mathcal{Q}_j$, $D_0(x) = 1/|\mathcal{X}|$ for every $x \in \mathcal{X}$, $Q_0(q) = 1/|\mathcal{Q}|$ for every $q \in \mathcal{Q}$,
$T = n \cdot \max\{\log|\mathcal{X}|, \log|\mathcal{Q}|\}, \quad \eta = \frac{\varepsilon}{2\sqrt{T\log(1/\delta)}}, \quad s = 12T$

DataPlayer:
  On input a query $\widehat{q}_t$, for each $x \in \mathcal{X}$:
    Update $D_t(x) = D_{t-1}(x) \cdot \exp\left(-\eta\left(\frac{1+\widehat{q}_t(D)-\widehat{q}_t(x)}{2}\right)\right)$
  Choose $\widehat{x}_t \leftarrow_{\mathrm{R}} \widetilde{D}_t$ and send $\widehat{x}_t$ to QueryPlayer

QueryPlayer:
  On input a data element $\widehat{x}_t$, for each $q \in \mathcal{Q}$:
    Update $Q_{t+1}(q) = Q_t(q) \cdot \exp\left(-\eta\left(\frac{1+q(D)-q(\widehat{x}_t)}{2}\right)\right)$
    Let $P_{t+1} = \Gamma_s Q_{t+1}$
  Choose $\widehat{q}_{t+1} \leftarrow_{\mathrm{R}} \widetilde{P}_{t+1}$ and send $\widehat{q}_{t+1}$ to DataPlayer

GenerateSynopsis:
  Let $\widehat{D} = (\widehat{x}_1, \ldots, \widehat{x}_T)$.
  Run sparse vector on $\widehat{D}$, obtain a set of at most $s$ queries $\mathcal{Q}_f$
  Run Laplace Mechanism, obtain answer $a_q$ for each $q \in \mathcal{Q}_f$
  Output $\widehat{D}$ to all analysts.
  For each $q \in \mathcal{Q}_f$, output $(q, a_q)$ to the analyst that issued $q$.
```

PROOF. Observe that the algorithm is computing an approximate equilibrium of the game $G_D(x,q) = \frac{1+q(D)-q(x)}{2}$. Let v, v_s be the value and constrained value of this game, respectively. First, we pin down the quantities v and v_s.

CLAIM 3.2. *For every D, the value and constrained value of G_D is $1/2$.*

We omit the proof, which considers the payoff to the data player if he plays the true database D as his strategy.

Let $\widehat{D} = \frac{1}{T}\sum_{t=1}^T x_t$. By Corollary 2.14,

$$v_s - 2\rho \leq \max_{Q\in\Delta_s(\mathcal{Q})}\left(\frac{1}{2}\mathop{\mathbb{E}}_{q\leftarrow_{\mathrm{R}}\widetilde{Q}}\left[1+q(D)-q(\widehat{D})\right]\right) \leq v + 2\rho.$$

Applying Claim 3.2 and rearranging terms, with probability at least $1-\beta/3$,

$$\left|\max_{Q\in\Delta_s(\mathcal{Q})}\left(\mathop{\mathbb{E}}_{q\leftarrow_{\mathrm{R}}\widetilde{Q}}\left[q(D)-q(\widehat{D})\right]\right)\right|$$
$$= O\left(\frac{\sqrt{\log(|\mathcal{X}|+|\mathcal{Q}|)\log(1/\delta)}+\log(1/\beta)}{\varepsilon\sqrt{n}}\right) := \alpha_{\widehat{D}}.$$

The previous statement suffices to show that $|q(D) - q(\widetilde{D})| \leq \alpha_{\widehat{D}}$ for all but s queries. Otherwise, the uniform distribution over the bad queries would be a distribution over queries contained in $\Delta_s(\mathcal{Q})$, with expected error larger than $\alpha_{\widehat{D}}$.

We can now run the sparse vector algorithm (Lemma 2.5). With probability at least $1-\beta/3$, it will identify every query q with error larger than $\alpha_{\widehat{D}} + \alpha_{SV}$ for

$$\alpha_{\mathrm{SV}} = O\left(\frac{\sqrt{s\log(1/\delta)}\log(|\mathcal{Q}|/\beta)}{\varepsilon n}\right).$$

Since there are at most s such queries, with probability at least $1-\beta/3$, the Laplace mechanism (Lemma 2.8) answers these queries to within error

$$\alpha_{\mathrm{Lap}} = O\left(\frac{\sqrt{s\log(1/\delta)}\log(s/\beta)}{\varepsilon n}\right).$$

Now, observe that in the final output, there are two ways that a query can be answered: either by $\widehat{D}$, in which case its answer can have error as large as $\alpha_{\widehat{D}} + \alpha_{\mathrm{SV}}$, or by the Laplace mechanism, in which case its answer can have error as large as α_{Lap}. Thus, with probability at least $1-\beta$, every query has error at most $\max\{\alpha_{\widehat{D}} + \alpha_{\mathrm{SV}}, \alpha_{\mathrm{Lap}}\}$. Substituting our choice of $s = 12T = O(n\log(|\mathcal{X}|+|\mathcal{Q}|))$ and simplifying, we conclude that the mechanism is (α,β)-accurate for

$$\alpha = O\left(\frac{\sqrt{\log(|\mathcal{X}|+|\mathcal{Q}|)\log(1/\delta)}\log(|\mathcal{Q}|/\beta)}{\varepsilon\sqrt{n}}\right).$$

□

Data Privacy

THEOREM 3.3. *Algorithm 3 satisfies (ε,δ)-differential privacy for the data.*

Before proving the theorem, we will state a useful lemma about the Bregman projection onto the set of high density measures (Definition 2.11).

LEMMA 3.4 (PROJECTION PRESERVES PRIVACY). *Let $A_0, A_1 : \mathcal{A} \to [0,1]$ be two full-support measures over a set of actions $\mathcal{A}$ and $s \in (0, |\mathcal{A}|)$ be such that $|A_0|, |A_1| \leq s$ and $|\ln(A_0(a)/A_1(a))| \leq \varepsilon$ for every $a \in \mathcal{A}$. Let $A_0' = \Gamma_s A_0$ and $A_1' = \Gamma_s A_1$. Then $|\ln(A_0'(a)/A_1'(a))| \leq 2\varepsilon$ for every $a \in \mathcal{A}$.*

We omit the proof of this lemma for lack of space. Now we prove the main result of this section.

PROOF OF THEOREM 3.3. We focus on analyzing the privacy properties of the output $\widehat{D} = (\widehat{x}_1, \ldots, \widehat{x}_T)$, the privacy of the final stage of the mechanism will follow from standard arguments in differential privacy. We will actually show the stronger guarantee that the sequence $v = (\widehat{x}_1, \widehat{q}_1, \ldots, \widehat{x}_T, \widehat{q}_T)$ is differentially private for the data. Fix a pair of adjacent databases $D_0 \sim D_1$ and let V_0, V_1 denote the distribution on sequences v when the mechanism is run on database D_0, D_1 respectively. We will show that with probability at least $1-\delta/3$ over $v = (\widehat{x}_1, \widehat{q}_1, \ldots, \widehat{x}_T, \widehat{q}_T) \leftarrow_{\mathrm{R}} V_0$, $|\ln(V_0(v)/V_1(v))| \leq \varepsilon/3$, which is no weaker than $(\varepsilon/3, \delta/3)$-differential privacy. To do so, we analyze the privacy of each element of v, $\widehat{x}_t$ or $\widehat{q}_t$, and apply the composition analysis of Dwork, Rothblum, and Vadhan [8]. Define $\varepsilon_0 = 2\eta T/n$.

CLAIM 3.5. *For every v, and every $t \in [T]$,*

$$\left|\ln\left(\frac{V_0(\widehat{x}_t \mid \widehat{x}_1, \widehat{q}_1, \ldots, \widehat{x}_{t-1}, \widehat{q}_{t-1})}{V_1(\widehat{x}_t \mid \widehat{x}_1, \widehat{q}_1, \ldots, \widehat{x}_{t-1}, \widehat{q}_{t-1})}\right)\right| \leq \varepsilon_0.$$

PROOF OF CLAIM 3.5. The left-hand side is as follows.

$$\left|\ln\left(\frac{\exp\left(-(\eta/2)\sum_{j=1}^{t-1} 1 + \widehat{q}_j(D_0) - \widehat{q}_j(\widehat{x}_t)\right)}{\exp\left(-(\eta/2)\sum_{j=1}^{t-1} 1 + \widehat{q}_j(D_1) - \widehat{q}_j(\widehat{x}_t)\right)}\right)\right|$$
$$= \frac{\eta}{2}\left|\sum_{j=1}^{t-1}\widehat{q}_j(D_0) - \widehat{q}_j(D_1)\right| \leq \frac{\eta(t-1)}{2n} \leq \frac{\eta T}{2n} \leq \varepsilon_0$$

□

Claim 3.6. *For every v, and every $t \in [T]$,*

$$\left|\ln\left(\frac{V_0(\widehat{q}_t \mid \widehat{x}_1, \widehat{q}_1, \ldots, \widehat{x}_t)}{V_1(\widehat{q}_t \mid \widehat{x}_1, \widehat{q}_1, \ldots, \widehat{x}_t)}\right)\right| \leq \varepsilon_0.$$

Proof of Claim 3.6. The sample $\widehat{q}_t$ is made according to $\widetilde{P}_t$, which is the distribution corresponding to the projected measure P_t. First we'll look at the unprojected measure Q_t. Observe that, for any database D and query q,

$$Q_t(q) = \exp\left(-(\eta/2)\sum_{j=1}^{t-1} 1 + q(D) - q(\widehat{x}_j)\right).$$

Thus, if $Q_0(q)$ is the measure we would have when database D_0 is the input, and $Q_1(q)$ is the measure we would have when database D_1 is the input, then

$$\left|\ln\left(\frac{Q_0(q)}{Q_1(q)}\right)\right| \leq \frac{\eta}{2}\left|\sum_{j=1}^{t-1} q_j(D_0) - q_j(D_1)\right| \leq \frac{\eta T}{2n},$$

for every $q \in \mathcal{Q}$. Given that Q_0 and Q_1 satisfy this condition, Lemma 3.4 guarantees that the projected measures satisfy

$$|\ln(P_0(q)/P_1(q))| \leq \eta T/n.$$

Finally, we note that if the above condition is satisfied for every $q \in \mathcal{Q}$, then the distributions $\widetilde{P_0}, \widetilde{P_1}$ satisfy

$$\left|\ln\left(\widetilde{P}_0(q)/\widetilde{P}_1(q)\right)\right| \leq 2\eta T/n \leq \varepsilon_0,$$

because the value of the normalizer also changes by at most a multiplicative factor of $e^{\pm\eta T/n}$. We observe that $V_b(\widehat{q}_t \mid \widehat{x}_1, \widehat{q}_1, \ldots, \widehat{x}_t) = \widetilde{P}_b(\widehat{q}_t)$ for $b \in \{0,1\}$, which completes the proof of the claim. □

Now, the composition lemma (Lemma 2.9) (for $2T$-fold composition) guarantees that with probability at least $1-\delta/3$,

$$|\ln(V_0(v)/V_1(v))| \leq \varepsilon_0\sqrt{4T\log(3/\delta)} + 4\varepsilon_0^2 T,$$

which is at most $\varepsilon/3$ by our choice of ε_0. This implies that $\widehat{D}$ is $(\varepsilon/3, \delta/3)$-differentially private.

We note that the sparse vector computation to find the s queries with large error is $(\varepsilon/3, \delta/3)$-differentially private, by our choice of parameters (Lemma 2.8), and the answers to the queries found by sparse vector are $(\varepsilon/3, \delta/3)$-differentially private for our choice of parameters (Lemma 2.5). The theorem follows from composition.

Query Privacy

Theorem 3.7. *Algorithm 3 satisfies (ε, δ)-one-query-to-many-analyst differential privacy.*

Before proving query privacy of Algorithm 3, we will state a useful composition lemma. The lemma is a generalization of the "secrecy of the sample lemma" [15, 8] to the interactive setting. Consider the following game:

- Fix an (ε, δ)-differentially private mechanism $\mathcal{A} : \mathcal{U}^* \to \mathcal{R}$ and a bit $b \in \{0,1\}$. Let $D_0 = \emptyset$.
- For $t = 1, \ldots, T$:
  - The (randomized) adversary $\mathcal{B}(y_1, \ldots, y_t; r)$ chooses two distributions B_t^0, B_t^1 such that $SD(B_t^0, B_t^1) \leq \sigma$.
  - Choose $x_t \leftarrow_{\text{R}} B_t^b$ and let $D_t = D_{t-1} \cup \{x_t\}$.
  - Choose $y_t \leftarrow_{\text{R}} \mathcal{A}(D_t)$.

For a fixed mechanism $\mathcal{A}$ and adversary $\mathcal{B}$, let V^0 be the distribution on $(y_1, \ldots, y_T)$ when $b = 0$ and V^1 be the distribution on $(y_1, \ldots, y_T)$ when $b = 1$.

Lemma 3.8. *If $\varepsilon \leq 1/2$ and $T\sigma \leq 1/12$, then with probability at least $1 - T\delta - \delta'$ over $y = (y_1, \ldots, y_T) \leftarrow_{\text{R}} V^0$,*

$$\left|\ln\left(V^0(y)/V^1(y)\right)\right| \leq \varepsilon(T\sigma)\sqrt{2T\log(1/\delta')} + 30\varepsilon^2(T\sigma)T.$$

We omit the proof of this lemma for lack of space.

We also need another lemma about the Bregman projection onto the set of high-density measures (Definition 2.11)

Lemma 3.9. *Let $A_0 : \mathcal{A} \to [0,1]$ and $A_1 : \mathcal{A} \cup \{a^*\} \to [0,1]$ be two full-support measures over their respective sets of actions and $s \in (0, |\mathcal{A}|)$ be such that 1) $|A_0|, |A_1| \leq s$ and 2) $A_0(a) = A_1(a)$ for every $a \in \mathcal{A}$. Let $A_0' = \Gamma_s A_0$ and $A_1' = \Gamma_s A_1$. Then $SD(\widetilde{A}_0', \widetilde{A}_1') \leq 1/s$.*

We omit the proof of this lemma for lack of space. Now we can prove one-query-to-many-analyst privacy.

Proof of Theorem 3.7. Fix a database D. Consider two adjacent query sets $\mathcal{Q}_0 \sim \mathcal{Q}_1$ and, without loss of generality assume $\mathcal{Q}_0 = \mathcal{Q}_1 \cup \{q^*\}$ and that $q^* \in \mathcal{Q}_{\text{id}}$ for some analyst id. We write the output to all analysts as $v = (\widehat{x}_1, \ldots, \widehat{x}_T, b_1, \ldots, b_{|\mathcal{Q}|}, a_1, \ldots, a_{|\mathcal{Q}|})$ where $\widehat{D} = \{\widehat{x}_1, \ldots, \widehat{x}_T\}$ is the database that is released to all analysts, $b_1, \ldots, b_{|\mathcal{Q}|}$ is a sequence of bits that indicates whether or not $q_j(\widehat{D})$ is close to $q_j(D)$, and $a_1, \ldots, a_{|\mathcal{Q}|}$ is a sequence of approximate answers to the queries $q_j(D)$ (or $\perp$, if $q_j(\widehat{D})$ is already accurate). We write $v_{-\text{id}}$ for the portion of v that excludes outputs specific to analyst id's queries. Let V_0, V_1 be the distribution on outputs when the query set is $\mathcal{Q}_0$ and $\mathcal{Q}_1$, respectively.

We analyze the three parts of v separately. First we show that $\widehat{D}$, which is shared among all analysts, satisfies analyst privacy.

Claim 3.10. *With probability at least $1 - \delta$ over the samples $\widehat{x}_1, \ldots, \widehat{x}_T \leftarrow_{\text{R}} V_0$,*

$$\left|\ln\left(\frac{V_0(\widehat{x}_1, \ldots, \widehat{x}_T)}{V_1(\widehat{x}_1, \ldots, \widehat{x}_T)}\right)\right| \leq \varepsilon.$$

Proof of Claim 3.10. To prove the claim, we show how the output $\widehat{x}_1, \ldots, \widehat{x}_T$ can be viewed as the output of an instantiation of the mechanism analyzed by Lemma 3.8. For every $t \in [T]$ and $\widehat{q}_1, \ldots, \widehat{q}_{t-1}$, we define the measure D_t over database items to be

$$D_t(x) = \exp\left(-(\eta/2)\sum_{j=1}^{t-1} 1 + \widehat{q}_j(D) - \widehat{q}_j(x)\right).$$

Notice that if we replace a single query $\widehat{q}_\ell$ with $\widehat{q}_\ell'$ and obtain the measure D_t', then for every $x \in \mathcal{X}$, $\left|\ln\left(\widetilde{D}_t(x)/\widetilde{D}_t'(x)\right)\right| \leq \eta$. Thus we can view $\widehat{x}_t$ as the output of an η-differentially private mechanism $\mathcal{A}_D(\widehat{q}_1, \ldots, \widehat{q}_{t-1})$, which fits into the framework of Lemma 3.8. (Here, $\widehat{x}_t$ plays the role of y_t and $\widehat{q}_1, \ldots, \widehat{q}_{t-1}$ plays the role of D_{t-1} in the description of the game, while the input database D is part of the description of $\mathcal{A}$).

Now, in order to apply Lemma 3.8, we need to argue the distribution on samples $\widehat{q}_t$ when the query set is $\mathcal{Q}_0$ is *statistically close* to the distribution on samples $\widehat{q}_t$ when the query set is $\mathcal{Q}_1$. Fix any $t \in [T]$ and let Q_0, Q_1 be the measure Q_t over queries maintained by the query player when the input query set is $\mathcal{Q}_0, \mathcal{Q}_1$, respectively. For $q \neq q^*$, we have

$$Q_0(q) = Q_1(q) = \exp\left(-(\eta/2)\sum_{j=1}^{t-1} 1 + q(D) - q(\widehat{x}_j)\right).$$

Additionally, we set $Q_0(q^*) = 0$ (for notational convenience), while $Q_1(q^*) \in (0, 1]$. Thus, if we let $P_0 = \Gamma_s Q_0$ and $P_1 = \Gamma_s Q_1$, we will have $SD(\widetilde{P}_0, \widetilde{P}_1) \leq 1/s$ by Lemma 3.9. Since the statistical distance is $1/s = 1/12T$, we can apply Lemma 3.8 to show that with probability at least $1 - \delta$,

$$\left| \ln \left(\frac{V(\widehat{x}_1, \ldots, \widehat{x}_T)}{V'(\widehat{x}_1, \ldots, \widehat{x}_T)} \right) \right| \leq \frac{\eta\sqrt{T \log(1/\delta)}}{8} + \frac{5\eta^2 T}{2} \leq \varepsilon.$$

□

Now that we have shown $\widehat{D}$ satisfies (ε, δ)-one-query-to-many-analyst differential privacy, it remains to show that the remainder of the output satisfies perfect one-query-to-many-analyst privacy. Recall from the proof of Theorem 3.1 that $\widehat{D}$ will be accurate for all but s queries. That is, if we let $\{f_j\}_{j \in [|\mathcal{Q}|]}$ consist of the functions $f_j(D) = |q_j(D) - q_j(\widehat{D})|$, then $|\{j \mid f_j(D) \geq \alpha\}| \leq s$, where α is chosen as in Theorem 3.1. By Lemma 2.8, the sparse vector algorithm will release bits $b_1, \ldots, b_{|\mathcal{Q}|}$ (the indicator vector of the subset of queries with large error) such that for every $j \in [|\mathcal{Q}|]$, the distribution on b_j does not depend on any function $f_{j'}$ for $j' \neq j$. Thus, if z_{-a} contains all the bits of $b_1, \ldots, b_{|\mathcal{Q}|}$ that do not correspond to queries in $\mathcal{Q}_a$, then the distribution of $z_{-\mathrm{id}}$ does not depend on the queries asked by analyst id, and thus $z_{-\mathrm{id}}$ is perfectly one-query-to-many analyst private. Finally, for each query q_j such that $b_j = 1$, the output to the owner of that query will include $a_j = q_j(D) + z_j$ where z_j is an independent sample from the Laplace distribution. These outputs do not depend on any other query, and thus are perfectly one-query-to-many analyst private. This completes the proof of the theorem.

4. A ONE-ANALYST-TO-MANY-ANALYST PRIVATE MECHANISM

In this section we present an algorithm for answering linear queries that satisfies the stronger notion of one-analyst-to-many-analyst privacy. The algorithm is similar to Algorithm 3, but with two notable modifications.

First, instead of the "query player" of Algorithm 3, we will have an "analyst player" who chooses analysts as actions and is trying to find an analyst id $\in [m]$ for which there is at least one query in $\mathcal{Q}_{\mathrm{id}}$ with large error (recall that the queries are given to the mechanism in sets $\mathcal{Q}_1, \ldots, \mathcal{Q}_m$). That is, the analyst player attempts to find id $\in [m]$ to maximize $\max_{q \in \mathcal{Q}_{\mathrm{id}}} q(D) - q(\widehat{D})$.

Second, we will compute a database $\widehat{D}$ such that $\max_{q \in \mathcal{Q}_{\mathrm{id}}} |q(D) - q(\widehat{D})|$ is small for all but s *analysts* in the set $[m]$, rather than having the s mishandled queries in Algorithm 3. We can still use sparse vector to find these s analysts, however we can't answer the queries with the Laplace mechanism, since each of the analysts may ask an exponential number of queries. However, since there are not too many analysts remaining, we can use s independent copies of the multiplicative weights mechanism (each run with $\varepsilon' \approx \varepsilon/\sqrt{s}$) to handle each analyst's queries. Due to space requirements we omit the proofs, which follow those of the previous section quite closely.

THEOREM 4.1. *Algorithm 4 is (α, β)-accurate for*

$$\alpha = \widetilde{O}\left(\frac{\sqrt{\log(|\mathcal{X}| + m) \log |\mathcal{Q}_{\mathrm{id}}| \log(m/\beta) \log^{3/4}(1/\delta)}}{\varepsilon n^{1/3}} \right).$$

THEOREM 4.2. *Algorithm 4 satisfies (ε, δ)-differential privacy for the data.*

THEOREM 4.3. *Algorithm 4 satisfies (ε, δ)-one-analyst-to-many-analyst differential privacy.*

Algorithm 4 Offline Mechanism for Linear Queries with One-Analyst-to-Many-Analyst Privacy

```
Input: Database $D \in \mathcal{X}^n$, and $m$ sets of linear queries $\overline{\mathcal{Q}}_1, \ldots, \overline{\mathcal{Q}}_m$. For id $\in [m]$, let $\mathcal{Q}_{\mathrm{id}} = \overline{\mathcal{Q}}_{\mathrm{id}} \cup \neg\overline{\mathcal{Q}}_{\mathrm{id}}$.
Initialize: Let $D_0(x) = 1/|\mathcal{X}|$ for each $x \in \mathcal{X}$, $I_0(q) = 1/m$ for each id $\in [m]$,

    $T = n^{2/3} \max\{\log |\mathcal{X}|, m\}, \quad \eta = \frac{\sqrt{T \log(1/\delta)}}{2\epsilon}, \quad s = 12T.$

DataPlayer:
  On input an analyst $\widehat{\mathrm{id}}_t$, for each $x \in \mathcal{X}$, update:
    $D_t(x) = D_{t-1}(x) \cdot \exp\left( -\eta \max_{q \in \mathcal{Q}_{\widehat{\mathrm{id}}_t}} \left( \frac{1 + \widehat{q}_t(D) - \widehat{q}_t(x)}{2} \right) \right)$
  Choose $\widehat{x}_t \leftarrow_{\mathrm{R}} \widetilde{D}_t$ and send $\widehat{x}_t$ to AnalystPlayer

AnalystPlayer:
  On input a data element $\widehat{x}_t$, for each id $\in \mathcal{I}$, update:
    $I_{t+1}(\mathrm{id}) = I_t(\mathrm{id}) \cdot \exp\left( -\eta \max_{q \in \mathcal{Q}_{\mathrm{id}}} \left( \frac{1 + q(D) - q(\widehat{x}_t)}{2} \right) \right)$
  Let $P_{t+1} = \Gamma_s I_{t+1}$
  Choose $\widehat{\mathrm{id}}_{t+1} \leftarrow_{\mathrm{R}} \widetilde{P}_{t+1}$ and send $\widehat{\mathrm{id}}_{t+1}$ to DataPlayer

GenerateSynopsis:
  Let $\widehat{D} = (\widehat{x}_1, \ldots, \widehat{x}_T)$
  Run sparse vector on $\widehat{D}$, obtain a set of at most $s$ analysts:
    $I_f = \{\mathrm{id}_1, \ldots, \mathrm{id}_s\} \subseteq [m]$
  For each analyst id $\in I_f$, run $\mathcal{A}_{\mathrm{MW}}(D, \mathcal{Q}_{\mathrm{id}})$ with parameters
    $\varepsilon' = \frac{\varepsilon}{10\sqrt{s \log(3s/\delta)}}$ and $\delta' = \frac{\delta}{3s}$
    Obtain a sequence of answers $\vec{a}_{\mathrm{id}}$.
  Output $\widehat{D}$ to all analysts.
  For each id $\in [m] \setminus I_f$, output $\vec{a}_{\mathrm{id}}$ to analyst id
```

5. A ONE-QUERY-TO-MANY-ANALYST PRIVATE ONLINE MECHANISM

In this section, we present a mechanism that provides one-query-to-many-analyst privacy in an online setting. The mechanism can give accurate answers to any *fixed* sequence of queries that are given to the mechanism one at a time, rather than the typical setting of *adaptively chosen* queries.

The mechanism is similar to the online multiplicative weights algorithm of Hardt and Rothblum [13]. In their algorithm, a hypothesis about the true database is maintained throughout the sequence of queries. When a query arrives, it is classified according to whether or not the current hypothesis accurately answers that query. If it does, then the query is answered according to the hypothesis. Otherwise, the query is answered with a noisy answer computed from the true database and the hypothesis is updated using the multiplicative weights update rule.

The main challenge in making that algorithm query private is to argue that the hypothesis does not depend too much on the previous queries. We overcome this difficulty by "sampling from the hypothesis." (recall that a database can be thought of as a distribution over the data universe). We must balance the need to take many samples — so that the database we obtain by sampling accurately reflects the hypothesis database, and the need to limit the impact of any one query on the sampled database. To handle both these constraints, we introduce *batching* — instead of updating every time we find a query not well-answered by the hypothesis, we batch together s queries at a time, and do one update on the average of these queries to limit the influence of any single query.

Algorithm 5 Analyst-Private Multiplicative Weights for Linear Queries

```
Input: Database $D \in \mathcal{X}^n$, sequence $q_1, \ldots, q_k$ of linear queries
Initialize: $D_0(x) = 1/|\mathcal{X}|$ for each $x \in \mathcal{X}$, $H_0 = D_0, \mathcal{U}_0 = \emptyset$,
$s_0 = s + \mathrm{Lap}(2/\varepsilon)$, $t = 0$, $r = 0$,

    $\eta = \frac{1}{n^{2/5}}, \quad s = \frac{128n^{2/5}\sqrt{\log|\mathcal{X}|\log(4k/\beta)\log(1/\delta)}}{\varepsilon},$

    $\widehat{n} = 32n^{4/5}\log(4k/\beta), \quad T = n^{4/5}\log|\mathcal{X}|, \quad R = 2sT,$

    $\sigma = \frac{20000\log^{3/4}|\mathcal{X}|\log^{1/4}(4k/\beta)\log^{5/4}(4/\delta)}{\varepsilon^{3/2}n^{2/5}},$

    $\tau = \frac{80000\log^{3/4}|\mathcal{X}|\log^{5/4}(4k/\beta)\log^{5/4}(4/\delta)}{\varepsilon^{3/2}n^{2/5}}.$

AnswerQueries:
While $t < T, r < R, i \leq k$, on input query $q_i$:
  Let $z_i = \mathrm{Lap}(\sigma)$
  If $|q_i(D) - q_i(H_t) + z_i| \leq \tau$: Output $q_i(H_t)$
  Else:
    Let $u = \mathrm{sgn}(q_i(H_t) - q_i(D) - z_i) \cdot q_i, \mathcal{U}_t = \mathcal{U}_t \cup \{u\}$
    Output $q_i(D) + z_i$
    Let $r = r + 1$
    If $|\mathcal{U}_t| > s_t$:
      Let $(D_{t+1}, H_{t+1}) = \mathbf{Update}(D_t, \mathcal{U}_t)$
      Let $\mathcal{U}_{t+1} = \emptyset$, $s_{t+1} = s + \mathrm{Lap}(2/\varepsilon)$
      Let $t = t + 1$
  Advance to query $q_{i+1}$

Update:
  Input: distribution $D_t$, update queries $\mathcal{U}_t = \{u_1, \ldots, u_{s_t}\}$
  For each $x \in \mathcal{X}$:
    Let $\mathbf{u}_t(x) = \frac{1}{3s}\sum_{j=1}^{s_t} u_j(x)$
    Update $D_{t+1}(x) = \exp(-(\alpha'/2)\mathbf{u}_t(x))D_t(x)$
  Normalize $D_{t+1}$
  Let $H_{t+1}$ be $\widehat{n}$ independent samples from $D_{t+1}$
  Return: $(D_{t+1}, H_{t+1})$
```

A note on terminology: the execution of the algorithm takes place in several rounds, where each round processes one query. Rounds where the query is answered using the real database are called *bad rounds*; rounds that are not bad are *good rounds*. We will split the rounds into T *epochs*, where the hypothesis H_t is used during epoch t.

Accuracy.

Accuracy. First, we sketch a proof that the online mechanism answers linear queries accurately. Intuitively, there are three ways that our algorithm might give an inaccurate answer, and we treat each separately. First, in a good round, the answer given by the hypothesis may be a bad approximation to the true answer. Second, in a bad round, the answer given may have too much noise. We address these two cases with straightforward arguments showing that the noise is not too large in any round.

The third way the algorithm may be inaccurate is if there are more than R bad rounds, and the algorithm terminates early. We show that this is not the case using a potential argument: after sufficiently many bad rounds, the hypothesis D_T and the sample H_T will be accurate for all queries in the stream, and thus there will be no more bad rounds. The potential argument is a simple extension of the argument in Hardt and Rothblum [13] that handles the additional error coming from taking samples from D_t to obtain H_t.

THEOREM 5.1. *Algorithm 5 is (α, β)-accurate for*

$$\alpha = O\left(\frac{\log^{3/2}(k/\beta)\sqrt{\log|\mathcal{X}|\log(1/\delta)}}{\varepsilon^{3/2}n^{2/5}}\right).$$

Due to space constraint, we omit the proof.

Data Privacy

THEOREM 5.2. *Algorithm 5 is (ε, δ)-differentially private.*

The proof follows from the modular privacy proof in [11].

Query Privacy. More interestingly, we show that this mechanism satisfies one-query-to-many-analyst privacy.

THEOREM 5.3. *Algorithm 5 is (ϵ, δ)-one-query-to-many-analyst private.*

PROOF. Fix the input database D and the coins of the Laplace noise — we will show that for every value of the Laplace random variables, the mechanism satisfies analyst privacy. Consider any two adjacent sequences of queries $\mathcal{Q}_0, \mathcal{Q}_1$. Without loss of generality, we will assume that $\mathcal{Q} = q_1, \ldots, q_k$ and $\mathcal{Q}' = q^*, q_1, \ldots, q_k$. For notational simplicity, we assume that every query in $\mathcal{Q}$ has a fixed index, regardless of the presence of q^*. More generally, we could identify each query in the sequence by a unique index (say, a long random string) that is independent of the other queries. We want to argue that the answers to *all queries in* $\mathcal{Q}$ are private, but *not* that the answer to q^* is private (if it is requested).

We will represent the answers to the queries in $\mathcal{Q}$ by a sequence $\{(H_t, i_t)\}_{t\in[T]}$ where H_t is the hypothesis used in the t-th epoch and i_t is the index of the last query in that epoch (the one that caused the mechanism to switch to hypothesis H_t). Observe that for a fixed database D, Laplace noise, and sequence of queries $\mathcal{Q}$, we can simulate the output of the mechanism *for all queries in* $\mathcal{Q}$ given only this information — once we fix a hypothesis H_t, we can determine whether any query q will be added to the update pool in this epoch. So once we begin epoch t with hypothesis H_t, we have fixed all the bad rounds, and once we are given i_t, we have determined when epoch t ends and epoch $t+1$ begins. At this point, we fix the next hypothesis H_{t+1} and continue simulating.

Formally, let V_0, V_1 be distribution over sequences $\{(H_t, i_t)\}$ when the query sequence is $\mathcal{Q}_0, \mathcal{Q}_1$, respectively. We will show that with probability at least $1-\delta$, if $\{(H_t, i_t)\}_{t\in[T]}$ is drawn from V_0, then $|\ln(V_0(\{(H_t, i_t)\})/V_1(\{(H_t, i_t)\}))| \leq \varepsilon$.

Recall that $\mathcal{U}_t$ is the set of queries that are used to update the distribution D_t to D_{t+1}. We will use $\mathcal{U}_{\leq t} = \bigcup_{j=0}^{t}\mathcal{U}_t$ to denote the set of all queries used to update the distributions $D_0, \ldots, D_t$. Notice that if q^* does not get added to the set $\mathcal{U}_0$, then V_0 and V_1 will be distributed identically. Therefore, suppose $q^* \in \mathcal{U}_0$. First we must reason about the joint distribution of the first component of the output. We omit the proofs.

CLAIM 5.4. *For all H_0, i_0, $|\ln(V_0(H_0, i_0)/V_1(H_0, i_0))| \le \varepsilon/2$.*

Now we reason about the remaining components $(H_1, i_1), \ldots, (H_T, i_T)$.

CLAIM 5.5. *For every H_0, i_0, with probability at least $1 - \delta$ over the choice of components $v = (H_1, i_1, \ldots, H_T, i_T) \leftarrow_R (V_0 \mid v_{t-1})$, we have $|\ln(V_0(v \mid H_0, i_0)/V_1(v \mid H_0, i_0))| \le \varepsilon/2$.*

Combining these two claims proves the theorem. □

Handling Arbitrary Low-Sensitivity Queries. We can also modify this mechanism to answer arbitrary Δ-sensitive queries, albeit with worse accuracy bounds. As with our offline algorithms, we modify the algorithm to run the multiplicative weights updates over the set of databases $\mathcal{X}^n$ and adjust the parameters. When we run multiplicative weights over a support of size $|\mathcal{X}|^n$ (rather than $|\mathcal{X}|$), the number of epochs increases by a factor of n, which in turn affects the amount of noise we have to add to ensure privacy. We omit this calculation for lack of space.

The final error bound we obtain (ignoring the parameters β and δ) is

$$O\left(\frac{\Delta^{2/5} n^{3/10} \log^{3/10} |\mathcal{X}| \log^{9/20} k}{\varepsilon^{2/5}}\right),$$

which gives a non-trivial error guarantee when $\Delta \ll 1/n^{3/4}$.

6. CONCLUSIONS

We have shown that it is possible to privately answer many queries while also preserving the privacy of the data analysts even if multiple analysts may collude, or if a single analyst may register multiple accounts with the data administrator. In the one-query-to-many-analyst privacy for linear queries in the non-interactive setting, we are able to recover the nearly optimal $\widetilde{O}(1/\sqrt{n})$ error bound achievable without promising analyst privacy. However, it remains unclear whether this bound is achievable for one-analyst-to-many-analyst privacy, or for non-linear queries, or in the interactive query release setting.

We have also introduced a novel view of the private query release problem as an equilibrium computation problem in a two-player zero-sum game. This allows us to encode different privacy guarantees by picking strategies of the different players and the neighboring relationship on game matrices (i.e., differing in a single row for analyst privacy, or differing by $1/n$ in ℓ_∞ norm for data privacy). We expect that this will be a useful point of view for other problems. In this direction, it is known how to privately compute equilibria in certain types of multi-player games [16]. Is there a useful way to use this multi-player generalization when solving problems in private data release, and what does it mean for privacy?

Acknowledgments. We thank the anonymous reviewers, along with Jake Abernethy and Salil Vadhan, for their helpful suggestions.

7. REFERENCES

[1] BARAK, B., HARDT, M., AND KALE, S. The Uniform Hardcore Lemma via Approximate Bregman Projections. In *SODA* (2009), pp. 1193–1200.

[2] BLUM, A., DWORK, C., MCSHERRY, F., AND NISSIM, K. Practical privacy: the sulq framework. In *PODS* (2005).

[3] BLUM, A., LIGETT, K., AND ROTH, A. A learning theory approach to non-interactive database privacy. In *STOC* (2008).

[4] DINUR, I., AND NISSIM, K. Revealing information while preserving privacy. In *PODS* (2003).

[5] DWORK, C., MCSHERRY, F., NISSIM, K., AND SMITH, A. Calibrating noise to sensitivity in private data analysis. In *TCC* (2006).

[6] DWORK, C., NAOR, M., REINGOLD, O., ROTHBLUM, G., AND VADHAN, S. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC* (2009).

[7] DWORK, C., NAOR, M., AND VADHAN, S. P. The Privacy of the Analyst and The Power of the State. In *FOCS* (2012).

[8] DWORK, C., ROTHBLUM, G. N., AND VADHAN, S. P. Boosting and Differential Privacy. In *FOCS* (2010), pp. 51–60.

[9] FREUND, Y., AND SCHAPIRE, R. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory* (1996), ACM, pp. 325–332.

[10] GUPTA, A., HARDT, M., ROTH, A., AND ULLMAN, J. Privately releasing conjunctions and the statistical query barrier. In *STOC* (2011).

[11] GUPTA, A., ROTH, A., AND ULLMAN, J. Iterative constructions and private data release. In *TCC* (2012).

[12] HARDT, M., LIGETT, K., AND MCSHERRY, F. A simple and practical algorithm for differentially private data release. *BIPS* (2012).

[13] HARDT, M., AND ROTHBLUM, G. N. A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis. In *FOCS* (2010), pp. 61–70.

[14] HSU, J., ROTH, A., AND ULLMAN, J. Differential privacy for the analyst via private equilibrium computation. *CoRR abs/1211.0877* (2012).

[15] KASIVISWANATHAN, S., LEE, H., NISSIM, K., RASKHODNIKOVA, S., AND SMITH, A. What can we learn privately? *SIAM Journal on Computing 40*, 3 (2011), 793–826.

[16] KEARNS, M., PAI, M., ROTH, A., AND ULLMAN, J. Mechanism design in large games: Incentives and privacy. *arXiv preprint arXiv:1207.4084* (2012).

[17] MCGREGOR, A., MIRONOV, I., PITASSI, T., REINGOLD, O., TALWAR, K., AND VADHAN, S. The limits of two-party differential privacy. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on* (2010), IEEE, pp. 81–90.

[18] MCSHERRY, F., AND TALWAR, K. Mechanism design via differential privacy. In *FOCS* (2007).

[19] RAKHLIN, A., AND SRIDHARAN, K. Statistical Learning Theory and Sequential Prediction. `http://www-stat.wharton.upenn.edu/~rakhlin/courses/stat928/stat928_notes.pdf`, 2012.

[20] ROTH, A. The algorithmic foundations of data privacy, course notes. `http://www.cis.upenn.edu/~aaroth/courses/privacyF11.html`, 2011.

[21] ROTH, A., AND ROUGHGARDEN, T. Interactive privacy via the median mechanism. In *STOC* (2010).

[22] ULLMAN, J. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. *CoRR abs/1207.6945* (2012).